AD-A047 480 HONEYWELL INC MINNEAPOLIS MINN SYSTEMS AND RESEARCH --ETC F/G 9/2
DISTRIBUTED DATA PROCESSING TECHNOLOGY. VOLUME IX. DDP RATIONAL--ETC(U)
SEP 77 C HUANG, L BEAN DASG60-76-C-0087
UNCLASSIFIED 77SRC73 NL

1 OF 1
ADA
047480

END
DATE
FILMED
1-78
DDC

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (WHEN DATA ENTERED)

REPORT DOCUMENTATION PAGE — READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER

2. GOV'T ACCESSION NUMBER

3. RECIPIENT'S CATALOG NUMBER

9

4. TITLE (AND SUBTITLE)

6 Distributed Data Processing Technology. Volume IX. DDP Rationale: The Program Experience Point of View.

5. TYPE OF REPORT/PERIOD COVERED

Final Report. October 1976 to October 1977,

6. PERFORMING ORG. REPORT NUMBER

14 77SRC73

7. AUTHOR(S)

10 C. Huang
L. Bean

8. CONTRACT OR GRANT NUMBER(S)

15 DASG60-76-C-0087

9. PERFORMING ORGANIZATIONS NAME/ADDRESS

Honeywell Systems and Research Center
2600 Ridgway Parkway
Minneapolis, Minnesota 55413

10. PROGRAM ELEMENT,PROJECT,TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME/ADDRESS

Ballistic Missile Defense
Advanced Technology Center
Huntsville, Alabama 35807

12. REPORT DATE

11 September 1977

13. NUMBER OF PAGES

91

14. MONITORING AGENCY NAME/ADDRESS (IF DIFFERENT FROM CONT. OFF.)

12 92p.

15. SECURITY CLASSIFICATION (OF THIS REPORT)

Unclassified

15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (OF THIS REPORT)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (OF THE ABSTRACT ENTERED IN BLOCK 20, IF DIFFERENT FROM REPORT)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (CONTINUE ON REVERSE SIDE IF NECESSARY AND IDENTIFY BY BLOCK NUMBER) (continued)

Distributed data processing (DDP)	Site defense system
Parallel processing	Air traffic control
Associative processing	Bussing
DDP payoffs	Modular interconnection

20. ABSTRACT (CONTINUE ON REVERSE SIDE IF NECESSARY AND IDENTIFY BY BLOCK NUMBER)

This volume discusses DDP rationale from the program experience point of view. The payoffs of DDP concept achieved in contract studies are described. The payoffs and research issues of DDP from the perspective of the real-world Site Defense Experience are also discussed.

HD-168 REV 11/74

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 55 IS OBSOLETE

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (WHEN DATA ENTERED)

402 349 LB

19. Key Words (concluded)

Decentralized control
T1 transmission line
Parallel-element processing ensemble (PEPE)
Communications multiplexer
Bulk filtering

FOREWORD

The research documented in this volume was conducted under Ballistic Missile Defense Advanced Technology Center contract no. DASG60-76-C-0087, entitled "Distributed Data Processing Technology." This work was performed by Honeywell Systems and Research Center, Minneapolis, Minnesota, under the direction of Mr. C. R. Vick, Director, Data Processing Directorate, Ballistic Missile Defense Advanced Technology Center. Mr. J. Scalf was the BMDATC project engineer for this contract; Ms. B. C. Stewart was the Honeywell/GRC program manager.

This report covers work from October 1976 to October 1977. Mr. C. Huang is the author of Section 2 of this volume which summarizes the payoffs of distributed data processing projects. (Honeywell has also performed work on distributed processing projects for BMDATC-P; the payoffs of DDP for those projects have been documented in the BMDATC-P final report and are not included here.)

Section 3 of this volume contains information prepared by a key former Site Defense Software manager on the payoffs and research issues of distributed processing from the perspective of Site Defense Experience. In addition, Section 3 contains real-world requirements which must be considered in DDP Design Theory development. Networks Inc. contributed to this section on behalf of Honeywell.

This document is Volume IX of the final report. Other volumes of the report are the following:*

*Volumes V, VI, the appendix to Volume VI, and a section of Volume VIII were prepared by General Research Corporation.

Volume I	-	Management Summary
Volume II	-	DDP Rationale: The Program Planning Point of View
Volume III	-	DDP Rationale: The Technology Point of View
Volume IV	-	Application of DDP Technology to BMD: Architectures and Algorithms
Volume V	-	Application of DDP Technology to BMD: DDP Subsystem Design Requirements
Volume VI	-	Application of DDP Technology to BMD: Impact on Current DP Subsystem Design and Development Technologies
Volume VII	-	Application of DDP Technology to BMD: Experiments
Volume VIII	-	Application of DDP Technology to BMD: Research Performance Measurement

CONTENTS

CONTENTS (continued)

CONTENTS (concluded)

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (concluded)

LIST OF TABLES

SECTION 1

INTRODUCTION

OVERVIEW

This volume contains results of supporting studies and research performed in two major areas. The study results were used as background and input information for the DDP Design Theory Research reported in Volumes II, III, and V of this report and as background and input information for the SETS and Software Engineering Impact Studies in Volume VI and Experiments in Volume VII. (In order to provide independent and unbiased information, the major contributors to this volume were not associated with the main Honeywell/GRC research team and were not familiar with the Phase I DDP Statement of Work; they were simply asked to document their organization's past experiences as related to distributed data processing.)

In Section 2, the payoffs of seven past DDP projects are compared and the results are quantified (wherever hard data are available). In Section 3, site defense experience is examined in terms of implications for distributed data processing. (One of the major benefits of the work of Section 3 was that the Honeywell/GRC research team was exposed to the real-world development issues which must be addressed by a BMD DDP Design Theory.)

SECTION 2

DDP PAYOFFS vs. CDP PAYOFFS: PROJECT EXPERIENCE

INTRODUCTION AND SUMMARY

Each of the seven studies described in this section was aimed at defining a special problem-driven architecture for a unique application. All of the systems described achieve one or more of the payoffs of distributed data processing. The payoffs of DDP which were evaluated in these studies are the following:

- Increased performance (throughput)
- Increased fault tolerance
- Increased reliability
- Functional modularity
- Physical modularity
- Lower hardware cost
- Reduced software cost
- Growth potential (expandability)
- Lower total system cost
- Reduced size, weight, and power consumption
- Processors can be physically distributed

Since each study was aimed at a specific application, not all of the payoffs listed were considered or sought in every study. However, each of the problem-driven architectures developed in the seven studies achieved a number of these payoffs. Table 1 summarizes the payoffs achieved in each study. These payoffs are based on a comparison between using the problem-driven distributed architecture approach and a conventional approach for the same application problem. The entries in the table contain one of the following four designations:

- Yes--means that (1) there are substantial data from the study results showing that the payoff (of the row) is achieved by the system in the project (of the column), or (2) the payoff is an obvious implication by the special architecture system in the project.
- Probably Yes--means that the payoff in question has not been specifically studied for the project in question, but the payoff is very likely to be achieved due to the special architecture of the system in the project.
- Unknown--means that no knowledge of this payoff was derived for the project.
- No--means that the payoff was not achieved for the project in question.

TABLE 1. PROBLEM-DRIVEN DISTRIBUTED ARCHITECTURES (PAYOFFS VERSUS PAST PROJECT STUDIES)

Payoff	Past Project Study						
	1 Parallel Processor for Augmenting ARTS III	2 Distributed Processor/ Memory	3 Feasibility Demonstration of Distributed Processions for SSCC	4 Evaluation of Associative Processor in Communication Multiplexing	5 Associative Buffer Concentrator	6 PEPE Implementation Study	7 ABMDA Bulk Filter Concept Definition
Increased performance	Yes	Yes	Probably Yes	Yes	Yes	Yes	Yes
Increased fault tolerance	Yes	Probably yes	Probably yes	Probably yes	Probably yes	Yes	Probably yes
Increased reliability	Yes	Probably yes	Probably yes	Probably yes	Probably yes	Yes	Probably yes
Functional modularity	Yes	Yes	Yes	Unknown	Unknown	Probably yes	Probably yes
Physical modularity	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Lower hardware cost	Yes	Probably yes	No	Unknown	Unknown	Probably yes	Unknown
Reduced software cost	Probably yes	Unknown	Yes	Unknown	Unknown	Probably yes	Probably yes
Growth potential	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Lower total system cost	Yes	Probably yes	Probably yes	Unknown	Probably yes	Probably yes	Probably yes
Reduced size, weight, power	Unknown	Probably yes	Unknown	Unknown	Unknown	Unknown	Unknown
Processors can be physically distributed	No	Yes (50 to 100 ft)	Yes (up to 2000 ft)	No	No	No	No

PARALLEL PROCESSOR FOR AUGMENTING THE ARTS III SYSTEM

The purpose of this study for the Department of Transportation was to establish the viability of the concept of augmenting the existing Automated Radar Terminal System III (ARTS III) with a parallel processor to obtain a cost-effective advantage in a future deployment. The expected increase in airport air traffic was the primary requirement driver. Two alternative approaches were considered for meeting the increased requirements. One approach was to add more copies of general-purpose-type processors currently used in the ARTS III system. The other approach was to off-load certain Air Traffic Control (ATC) functions to a group of simple homogeneous processing elements (PEs) which operate in a parallel and associative fashion.

The specific objectives of this study program were:

- Develop the ATC requirements base relevant to current and future airport terminal environment.
- Develop a failsafe/failsoft hardware and software system configuration for the second approach (i.e., augmenting existing ARTS III with a parallel processor).
- Perform cost-effectiveness analysis comparing the first approach (i.e., using ARTS III general-purpose processors only) and the second approach.
- Provide a complete description of the proposed system configuration and a cost proposal for implementing this system in the future.

The results of this study indicated that the second approach was clearly better with respect to cost effectiveness, reliability, fault tolerance, and growth potential.

Of the nine major ATC functions, four were suitable for the parallel (or associative) processing:

- Beacon Input Processing
- Radar Target Detection
- Correlation and Tracking
- Conflict Prediction

In the second approach, these functions were off-loaded to the parallel processor. During the analysis, exact proven algorithms for these functions were applied to both approaches, so that there were no performance accuracy differences between the two approaches.

A summary of the cost-effectiveness analysis of the two approaches is shown in Figure 1. (Data shown are extracted from pages 8-68 through 8-76 of the contract final report). The system complexity ranges from 250 tracks (aircraft) to 1000 tracks. In all cases, the second approach is better. The cost is based on standard small-scale integration/medium-scale integration (SSI/MSI) implementations. The cost difference increases as the system complexity increases. Also shown in the figure is the number of PEs for each case in the second approach.

Figure 2 is a summary of the reliability analysis comparing the two approaches. (Data are extracted from pages 5-17 through 5-19 of the contract final report.) The mean up-time for the second approach is better than that of the first approach by a factor of 10 or more for each complexity level of system

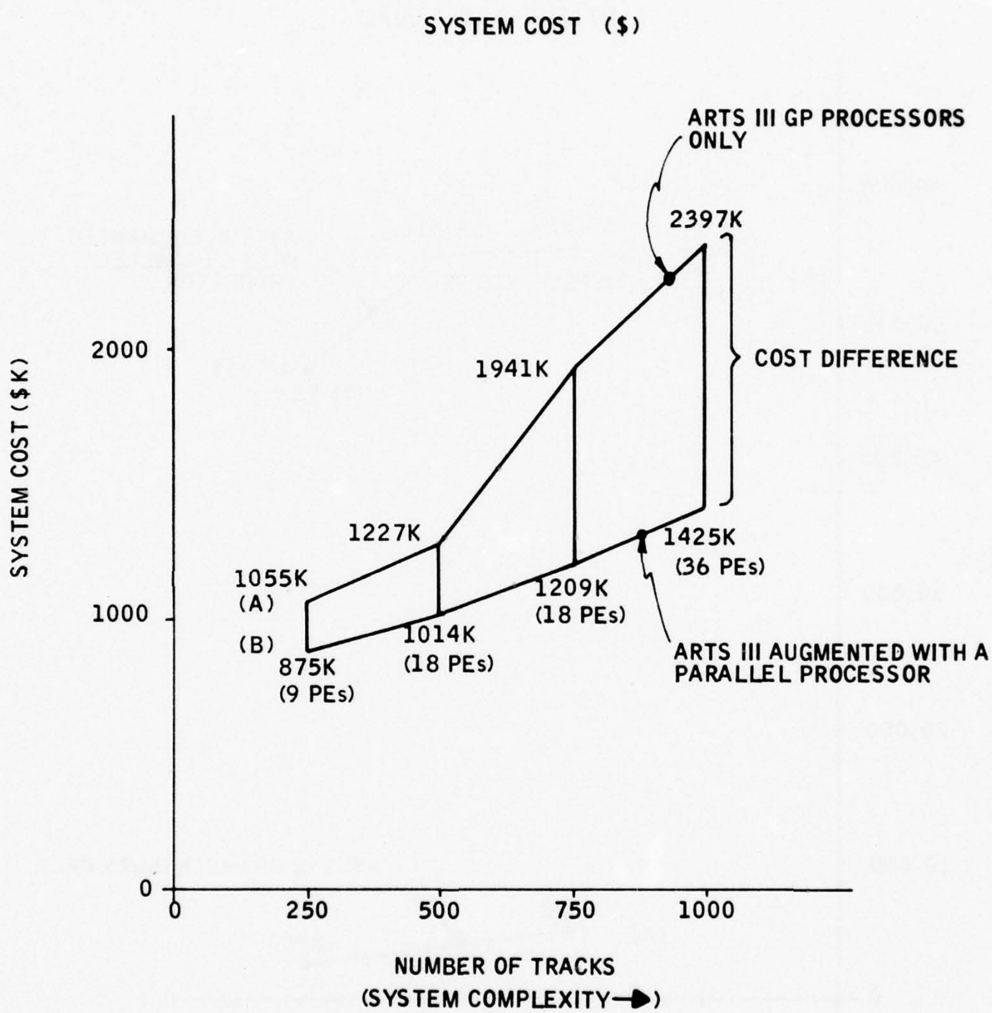


Figure 1. Cost Comparison Between Two Alternate Approaches

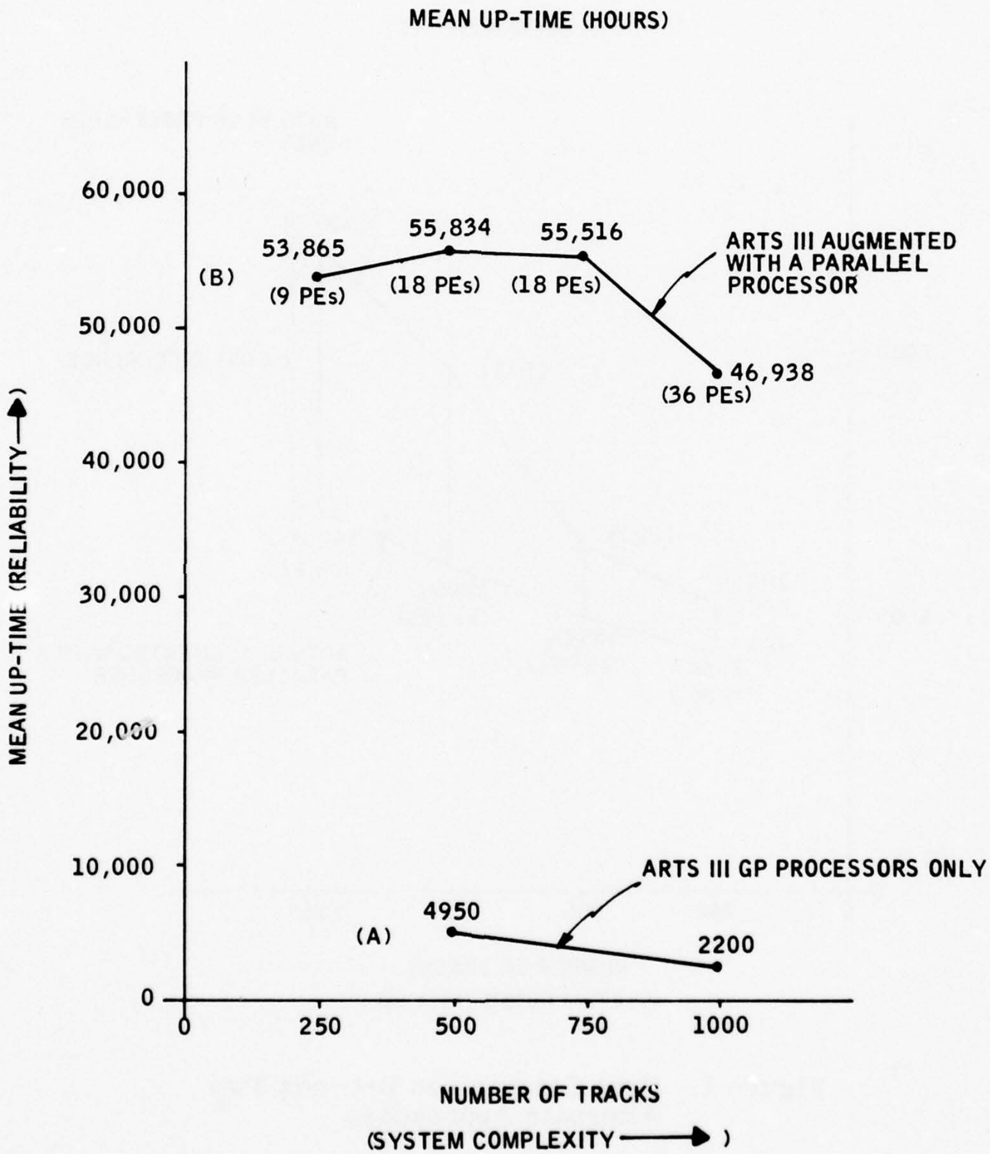


Figure 2. Mean Up-Time (Reliability) Comparison Between Two Alternate Approaches

configuration. The reliability analysis also concluded that (page 5-19 of contract final report) using the second approach, the mean up-time and the mean down-time exceed the production system requirements (specified by DOT) by a factor of 10 or more and a factor of two or more, respectively.

Using the parallel approach, fault tolerance could be achieved by using dual-redundant control units and by providing redundant (spare) PEs beyond those required for processing the function load. (For example, four of the 36 PEs used in the 1000-track system were spares.) The overall hardware redundancy needed in the parallel processor was only 30 percent, which provided the full redundancy needed for the failsafe/failsoft requirements.

The second approach with its modular processing elements was also suitable for system expansion to adapt to the growth of the future requirements. The system capability could be increased for more traffic load and/or more ATC functions (e.g., *Conflict Resolution*) with no changes to the system hardware and software other than addition of PEs and the specific application software programs.

This same study program was performed simultaneously and independently by three different companies: Texas Instruments, Goodyear Aerospace, and Honeywell Inc. The three companies concluded with similar results, that augmenting the ARTS III system with a parallel or associative processor was the more advantageous approach.

ALL-SEMICONDUCTOR DISTRIBUTED AEROSPACE PROCESSOR/MEMORY (DP/M) STUDY

The primary objective of the DP/M project for the Air Force Systems Command was to develop preliminary designs for an all-semiconductor DP/M system and its network interconnection pattern and to determine the

capabilities and limitations of this processing architecture for avionics applications. The specific goals for such a system included the following:

- The DP/M system should allow distribution of the computing workload for avionics among a number of simultaneously operating, interconnected elements.
- Since software costs are known to far exceed hardware costs for avionics computer systems, the software considerations would be of primary importance. The programmability of such a system should be carefully investigated.
- The design should allow for incremental expansion of system capability in a cost-effective manner.
- Physical distribution of PEs within the aircraft must be simple, low cost, and capable of operating over relatively long distances (e.g., 50 to 100 ft.).

At the start of this project, the potential advantages and problems of a distributed-processing system were considered. They are discussed briefly, next.

Potential Advantages

- Expandable-- The general architecture and operational characteristics could remain constant over a wide range of system sizes.
- Modular--By using a small PE as the basic building block, a wide variety of system sizes would be possible.

- Fault Tolerant--A distributed system using a number of identical elements and a common bus structure would be amenable to fault-tolerant techniques.

- All-Semiconductor Implementation--The distributed system could be implemented using large-scale integrated-circuit (LSIC) technology which has advantages on the size, weight, power, and cost.

Potential Problems

- Software--The fragmentation of programs in such a system could result in an unmanageable system from the standpoints of coding, verification, and maintenance of the software.

- Executive--Real-time control of a distributed system could result in an excessive overhead for the Executive Control function.

- Bussing/Interconnection--Data transfer between elements of the system via a bussing structure could become a significant factor in the total throughput of the system.

- Technology Limitations--Unless an individual PE can be implemented on one (or a few) LSI circuits, the system concept could be impractical in terms of reliability, logistics, and development costs.

Since the extent of these problems largely depends on the amenability of the avionics functions to partitioning, the processing requirements were first analyzed from the standpoint of computational positioning. Following the function paritioning effort, a preliminary design task was performed

focusing on the software, Executive system, bussing structure, and PE hardware of candidate distributed architectures. Two potential candidates were selected and designed in more detail. Finally, the technology alternatives to determine the feasibility of implementing the two final candidates were assessed.

This study concluded that there were sufficient data to indicate that the concept of a DP/M system was not only feasible, but also a potentially cost-effective approach to a total avionics system. Some specific results in support of this statement are:

- Based on an extensive analysis of computational requirements, the avionics problems are amenable to partitioning into subfunctions which "fit" within a small PE. Each PE contains a processor, its memory, and interfaces to other PEs and (I/O) devices (see Figure 3). The distributed system is an interconnection of homogeneous PEs. The complete system can be physically distributed within the aircraft.

- The software for a DP/M system does not appear to be a major risk area, so long as an adequate support software system is provided. Most languages used in the Air Force (SPL, JOVIAL, etc.) could be modified slightly and used for programming a DP/M system.

- The Executive Control requirements for a distributed architecture were found to be considerable less than originally anticipated, and this aspect of a DP/M system could be an advantage over some other architectures such as multiprocessors. This occurs because the location of programs and data is relatively fixed, and a processor is always

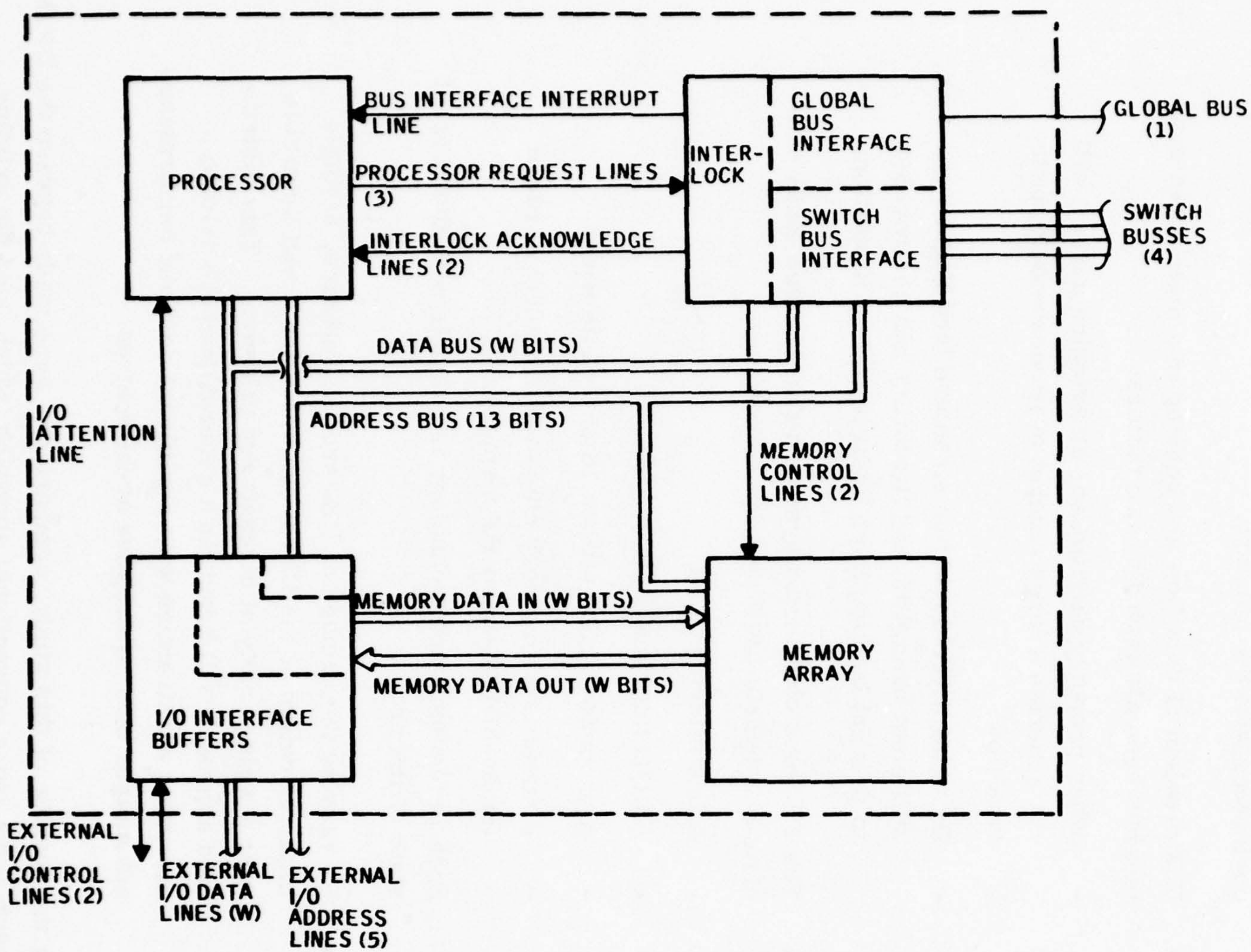


Figure 3. Distributed-Processor/Memory Element

available to execute the program, since the processor is co-located with the memory.

- The problem of data flow and bussing in a distributed system has been alleviated by two factors:
  - When properly partitioned, an avionics system will not generate a large amount of process-to-process data flow.
  - By using a two-level bus structure (see Figure 4), sufficient bandwidth and flexibility can be provided to prevent bussing from being a system limitation.
- The PE for a distributed system defined in this study consists of a hybrid LSIC package containing:
  - A bus interface LSIC
  - An I/O interface LSIC
  - A processor LSIC (either 16 or 24 bits wide)
  - Four (or six) memory chips to implement a 16-bit (or 24-bit) 4K-word PE memory.

  Such implementation technology is possible with the current state of the art.
- By varying the number of PEs and the software, a single implementation of the DP/M system concept will be usable over a wide variety of aircraft and missions. This distributed approach will result in a general-purpose avionic computer architecture with significant cost and performance advantages over alternative architectures.

Using the results of this study, comparisons were made between the DP/M approach and a more conventional approach (which uses the existing,

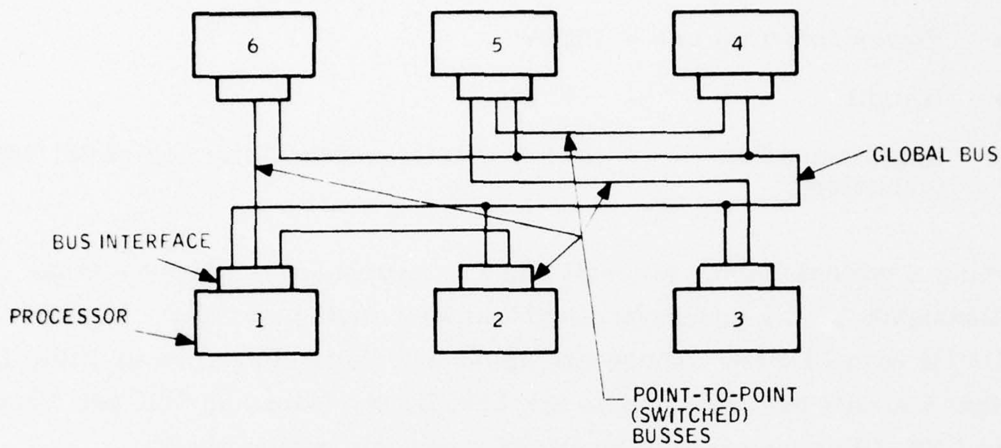


Figure 4. Hybrid Interconnection Scheme

more-powerful general-purpose computers) for handling the future avionic processing functions. The typical PE used in the DP/M approach will have the following characteristics:

- Throughput - 250 kips
- Size - 3 in^3
- Power consumption - 5 W
- Weight - 0.25 lb
- Implementation technology - LSI

The existing, conventional, and more-powerful general-purpose computer chosen for comparison is the UYK-19, which has the following characteristics:

- Throughput - 400 kips
- Size - 0.7 ft^3
- Power consumption - 200 W
- Weight - 38 lb
- Implementation technology - TTL (transistor-transistor logic) SSI/MSI

Assuming a typical fighter aircraft has a requirement for about 6 Mips total throughput,* the processing load can be handled by either 15 UYK-19 or a DP/M with 24 PEs. These two approaches are compared in Table 1. Note that the cost per throughputs are $56.7K per Mips and $8K per Mips for the UYK-19 approach and the DP/M approach, respectively.

TABLE 2. COMPARISON BETWEEN UYK-19 PROCESSORS AND DP/M APPROACH FOR A FIGHTER AIRCRAFT PROCESSING

Parameter	Approach	
	UYK-19	KP/M
No. of processors	15	24
Total throughput (Mips)	6	6
Implementation technology	TTL MSI/SSI	LSI
Total size (ft^3)	10	0.04
Power consumption (W)	3000	120
Total weight (lb)	570	6
Total cost ($K)	$340	$48
Cost/throughput ($K/Mips)	$56.7	$8

*This comparison is very rough and does not consider such things as system overhead differences.

FEASIBILITY DEMONSTRATION OF DISTRIBUTED PROCESSING FOR SMALL SHIPS COMMAND AND CONTROL (SSCC) SYSTEMS

The objective of this study for the Naval Electronic Laboratory Center was to develop and demonstrate a laboratory distributed-processing system oriented toward the requirements of SSCC data processing. A system, the Honeywell Experimental Distributed Processor (HXDP), was designed, built, and checked out. The system was demonstrated at NELC in May 1976.

The primary goals of this distributed-processing system were to reduce the real-time data-processing costs and to provide larger processing capability under the SSCC environments.

The general trend is that the hardware cost is decreasing while the software cost is increasing, and, gradually, the software cost has become the major system cost factor. It is increasingly feasible to use hardware to reduce the software costs.

The HXDP system was designed with the philosophy that it must be highly modular, have nearly all communications overhead handled in hardware, and be capable of being physically dispersed.

The HXDP system consists of a number of homogeneous processors. (See Figure 5.) Each processor consists of an off-the-shelf minicomputer with memory and a highly capable bus interface unit (BIU). Each BIU is a stored-program device (hardware/firmware) which manages and enforces Interprocess Communications protocols in the system. System control is completely decentralized so that there is no single entity which must manage the system.

The HXDP system incorporates a number of features which are important for SSCC processing development work:

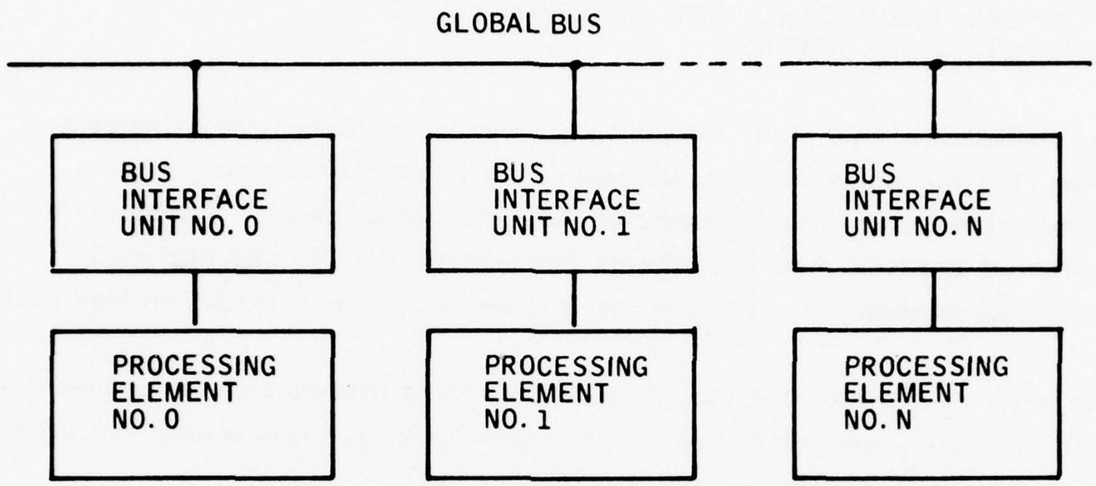


Figure 5. HXDP System Configuration

- Modular Interconnection -- The interconnection scheme is independent of the number of processors in the application system. A new function is added by connecting the new processor or processors at any point on the existing communication path.

- Communication Overhead Handled by Hardware -- The HXDP interconnection units provide the services required to move messages between computers with minimal software involvement. Management of the distributed-processing transmit/receive overhead is totally within the interconnect mechanism, freeing the computer to handle the real-time control applications.

- Decentralized Control -- The system communication mechanism is completely decentralized; there is no overall

controller. The communication mechanism cannot be disabled by the failure of any individual processor, thus inherently supporting graceful degradation in the presence of faults.

- Can be Physically Distributed -- The HXDP communication path can be up to 2000 feet long, allowing PEs to be physically located where they are needed onboard the ship.

EVALUATION OF THE USE OF AN ASSOCIATIVE PROCESSOR IN COMMUNICATION MULTIPLEXING

The objective of this study for Rome Air Development Center was to evaluate the potential use of associative-processing techniques in communications applications. Particular emphasis was placed on asynchronous multiplexing and concentration of voice and data signals for applications to a wideband data channel (e.g., T1 transmission line). The use of associative processing will permit multiplexing of signals which are not under direct clock control of the multiplexer. The goal was to develop an approach for improving the bandwidth utilization of wideband channel while eliminating other multiplexer and concentrator inefficiencies commonly found in current network applications. These deficiencies include hardware inflexibility, lack of expandability and economical growth capability, synchronization problems, inability to match the discontinuous and "bursty" input traffic flow to the available output channel capacity, and lack of capability to gracefully degrade operation during peak overload conditions.

The approach taken was to investigate asynchronous multiplexing, data concentration, voice-to-digital conversion, and associative processing. Various voice redundancy-removal algorithms were investigated and their suitability for implementation on an associative processor evaluated. An Associative Communication Multiplexer (ACM) incorporating an associative

architecture suitable for implementation of time-domain redundancy-reduction algorithms was designed and evaluated. A simplified ACM system block diagram is shown in Figure 6. A computer simulation was conducted and speech quality comparisons with actual voice processed by several candidate algorithms were made.

This study concluded that the use of the ACM concept and design can provide several substantial improvements over existing network equipment in the compression of voice, concentration of data, and multiplexing of both onto a high-speed transmission facility. These improvements include:

- Lower data rates for digitized "toll quality" voice -- about 4:1 reduction on bandwidth requirement
- Cost economies through combining voice, synchronous, and asynchronous digital data into a single facility
- Flexible priority structure for selecting only the most significant data from all input types (voice, data, etc.) to be asynchronously multiplexed onto a fixed-capacity, high-speed transmission facility
- Modular hardware structure for ease of expansion. Adding more inputs requires simple addition of PEs and I/O units with no software modification.
- All-digital for reliability
- Suitable for LSI, resulting in lower cost with volume usage

The information derived in this study formed the basis for an experimental evaluation of the ACM concept by using an experimental model and performing extensive listening tests. (The follow-on of this study was the Associative Buffer Concentrator study which is discussed next.)

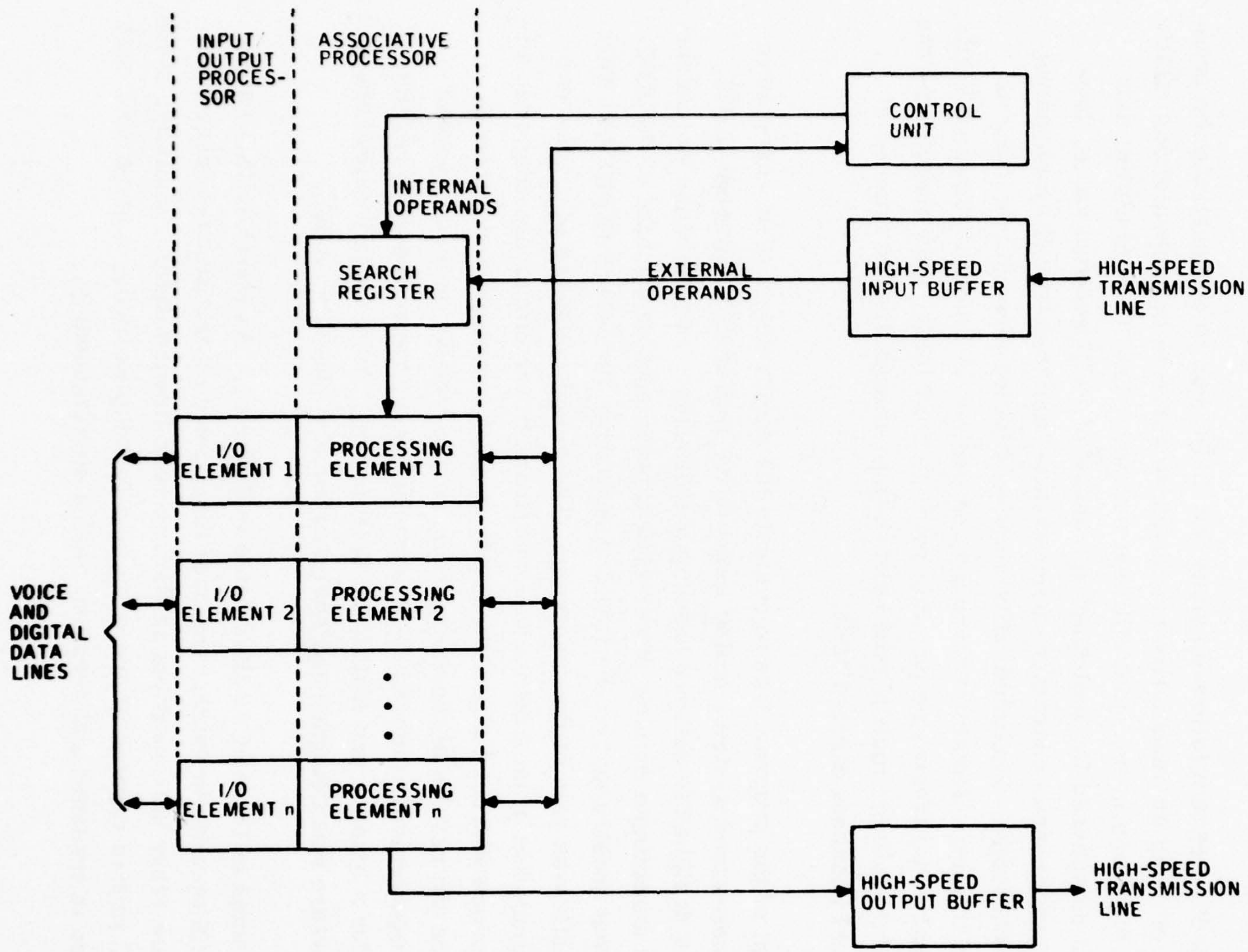


Figure 6. Simplified System Architecture for the ACM

ASSOCIATIVE BUFFER CONCENTRATOR (ABC)

The objective of this follow-on study for RADC was to demonstrate the practicality of using an associative processor to perform communications multiplexing functions in the most efficient manner. The types of digital data streams considered for multiplexing consist of PCM encoded voice, low-speed asynchronous character-oriented data, and high-speed synchronous bit-oriented data. Maximum efficiency is obtained through removing as much of the time-domain redundancy and periods of inactivity present in the individual data streams as possible prior to combining, thus maximizing the use of the available multiplexed bandwidth by increasing the number of individual channels carried on it.

The goal of the program was to evaluate the performance of an associative processor-based system, called Associative Buffer Concentrator (ABC), that was designed to achieve maximum efficiency. Performance was evaluated by executing a number of selected experiments on a model of the ABC which was constructed on the RADC Associative Processor (RADCAP) Test Bed Facility as part of this program. The model consisted of a non-real-time simulation of an ABC multiplexer/transmitter communicating over an error-prone wideband digital link with an ABC demultiplexer/receiver. Based on the results of these experiments, as well as on an analysis of other requirements for a first-level multiplexer, the processing requirements for a stand-alone ABC were determined. An associative-processor architecture was designed and analyzed for a stand-alone ABC.

The concept of the ABC is illustrated in Figure 7. As shown in this figure, the ABCs operate in pairs. One unit functions as a transmitter (receiver), while the other unit operates as the receiver (transmitter). Actually, since an ABC pair is capable of operating in a full duplex mode, a given ABC unit operates in transmit and receives modes simultaneously.

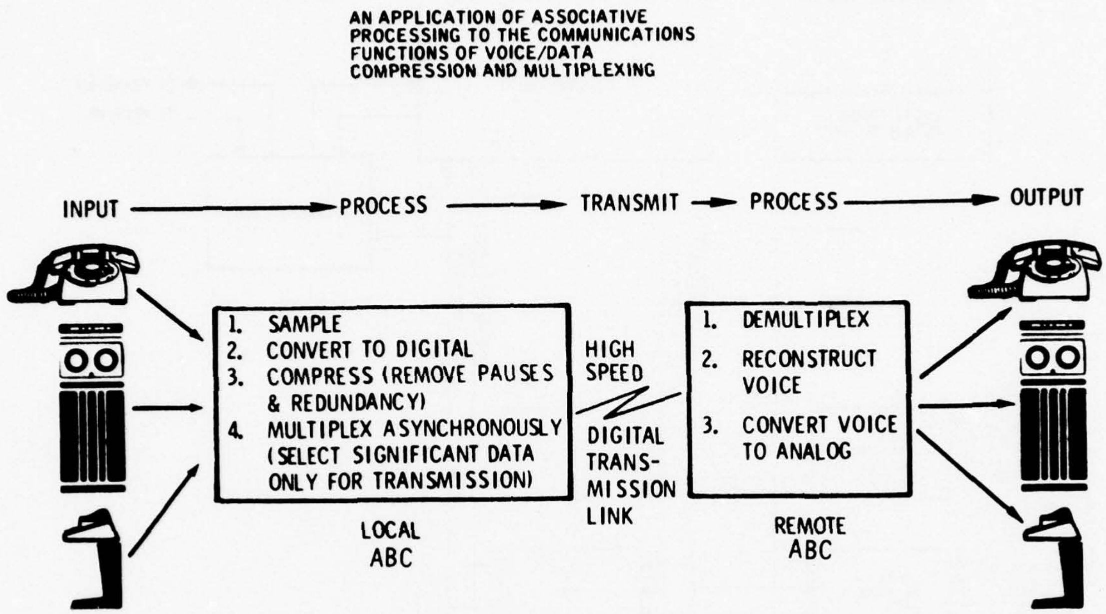


Figure 7. Associative Buffer Concentrator (ABC)

Figure 8 is a simplified architecture of an ABC unit. The architecture contains a highly modular ensemble of nearly identical computing elements. Although several versions of the I/O circuitry are necessary to interface the various analog and digital sources, each PE is identical.

The ABC uses the capabilities of associative processing in five primary areas:

- Parallel digitization and loading of input data from multiple channels
- Parallel compression and concentration of multiple input channels
- Associative search operations to select the most significant data for transmission

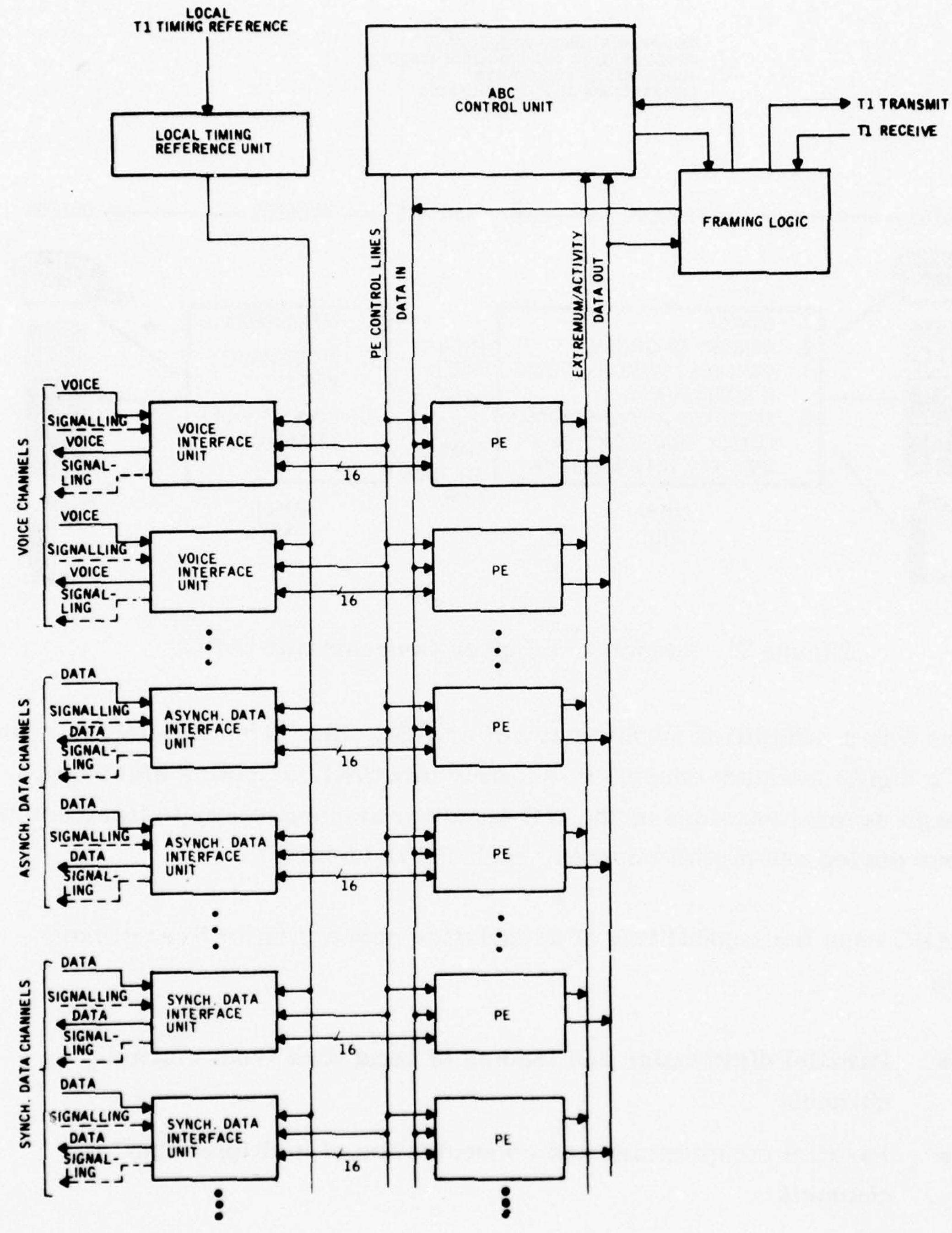


Figure 8. Overall ABC Block Diagram

- Parallel sample reconstruction for voice channels
- Parallel output data to multiple channels and D/A conversion for voice

The most important experimental results of this study program were intelligibility test scores for voice under extensive test conditions. One major conclusion drawn from the experimental results was that a system of up to 96 voice channels (each with a data rate of 64 kbps) can operate over a T1 line (bandwidth 1.544 Mbps) with at least 80 percent intelligibility (can be as high as 90 percent) and good quality under an error-prone environment with bit error rates to 10^{-3}. (The test scores were based on objective listening tests using Diagnostic Rhyme Tests for intelligibility.) An individual channel thus generates on the average a 16-kbps data stream, including overhead, and hence affords a 4:1 bandwidth utilization improvement factor over current, uncompressed PCM encoders. These results agree with the preliminary results obtained in the previous study phase. Figure 9 illustrates the improvement of this transmission line bandwidth utilization realized by using the ABC approach.

Because of the parallel feature of the operations for the voice compression and the multiplexing function and the very high-speed requirement of these operations (the ABC cycle time for parallel operations needs to be below 200 nsec), it is just simply impossible to use a conventional general-purpose computer to perform the ABC equivalent operations. (The conventional general-purpose computer would need a cycle time of about 2 nsec to replace an ABC system with 100 PEs. This is clearly beyond the current state of the art.)

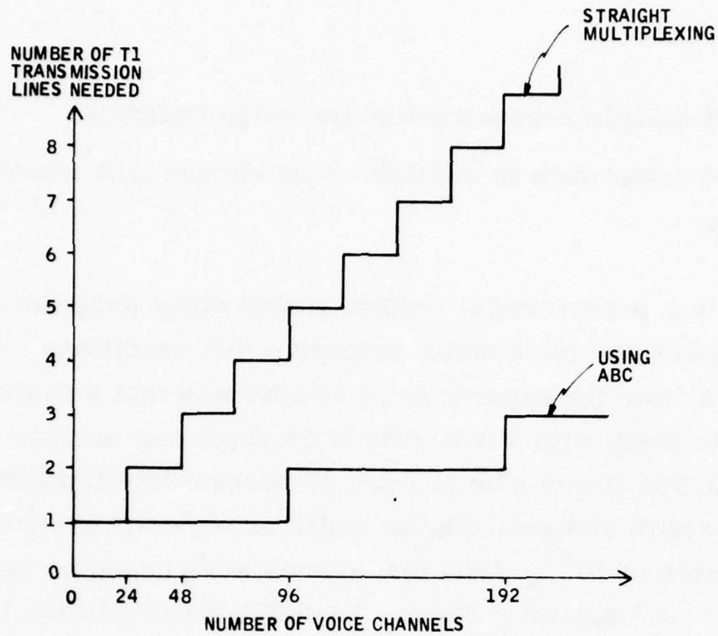


Figure 9. Improvement of Transmission Bandwidth Utilization Through Using the ABC Approach

PEPE IMPLEMENTATION STUDY

A Parallel-Element Processing Ensemble (PEPE) is a programmable special-purpose computing machine developed for the Advanced Ballistic Missile Defense Agency. It was designed to augment conventional sequential computers in ballistic missile defense (BMD) data processing. Combining associative- and parallel-processing techniques, it offers a promising approach to the vast data-processing requirements of BMD.

PEPE is termed an ensemble because it is an unstructured, indefinite number of processing elements which operate in parallel under global control with no direct communication between elements. A block diagram of the PEPE system is shown in Figure 10. Each element consists of three functional entities: an arithmetic unit, a correlation unit, and a memory. The control unit consists of two functional entities: an arithmetic control unit and a correlation control unit.

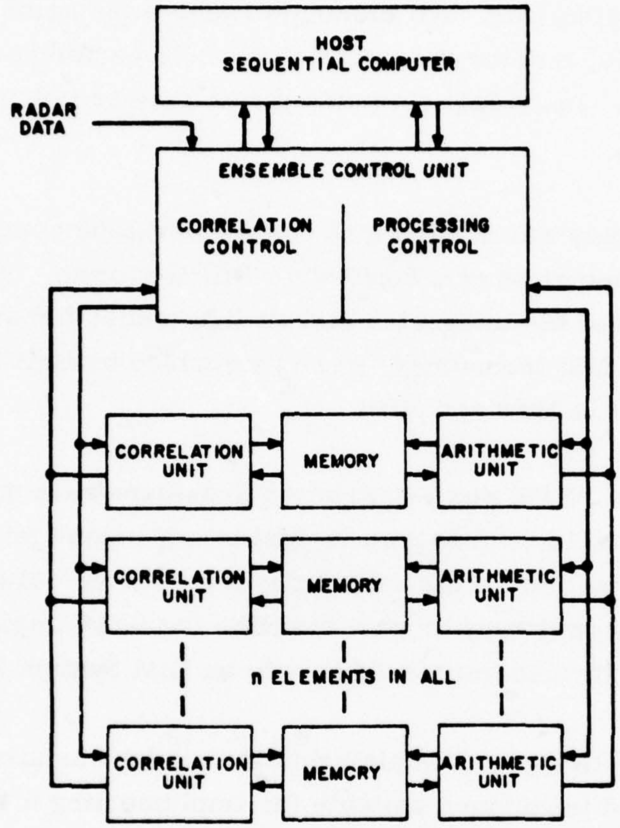


Figure 10. Block Diagram of
PEPE System

Program information is stored in the sequential (host) computer and instructions are supplied to the ensemble in the proper order. The number of elements (n) is made large enough to equal or exceed the size of the data base that must be handled. This degree of parallelism permits each object being observed by the radar to be assigned to an individual PE. The PEPE is then capable of operating on all objects simultaneously. With this approach, the processing time is no longer a strong function of the number of objects being observed. The global control units are used to control the operation of the

entire ensemble. Common input, output, and control busses go from the control units to all elements. All elements receive the same inputs and the same control signals, and the ensemble as a whole performs one common operation at a time. Individual elements participate or not, depending on their internal state.

This subcontract study was a third step toward development of hardware suitable for implementation of a deployable PEPE system. The first two steps, development of the integrated circuit (IC) model and an implementation study based on MSI technology, were performed by Bell Telephone Laboratories (BTL) in 1969 and 1970.

The IC model of the PEPE was constructed to demonstrate that the organization was in fact feasible, and to gain insight to any unique problems associated with the organization. This model was a research tool and not intended to serve as either a prototype or as a baseline for later implementations. This model had performed successfully with an IBM System 360/65 as a host.

This third step (the Honeywell implementation study) was aimed at identifying technologies and techniques suitable for implementing a PEPE system in the 1972 to 1977 time frame. The preliminary system design of a PEPE system was also performed during this subcontract study.

The results of this study showed that a reliable, cost-effective PEPE system could be implemented with 1972 hardware. The implementation techniques considered were amenable to incorporation of advances in technology both at the circuit packaging level and throughout the higher-level mechanical packaging.

Some novel features of the PEPE system are summarized as follows:

- Fully Parallel -- For the BMD application, the parallel-processing capability of PEPE significantly increases the computing capacity by processing multiple instances of data through a single operation. In general, an individual element is assigned to each object being observed by the radar. The ensemble can then operate on all active elements simultaneously, and thus process the common attributes of each object in a parallel mode.

- Associative -- The ensemble is addressed by content rather than by location, making the accessing and handling of data more direct, natural, and efficient. The elements are addressed on the basis of the attributes of the objects whose data they store. This feature contributes to making the element allocation simple and the reliability high.

- Growable -- Traffic growth can be achieved in a PEPE ensemble by adding elements. This growth can be made without impact on the software.

- Reliable -- The unstructured, highly parallel, and associative form of PEPE offers some interesting properties for system availability and reliability. There is obviously a high degree of redundancy. This is complemented by the lack of structure and avoidance of dependence on location which permit the use of an ensemble having a number of elements somewhat larger than required to meet the traffic requirements in anticipation of element failure. As element failure is detected, the offending element is simply decommissioned electronically. Since element location is of no consequence, a lost element need not be replaced immediately but may remain in place until replacement and repair are convenient.

- Realizable -- The computational gains realized with PEPE are achieved by architectural innovations, not hardware performance advances. In fact, the circuit performance requirements are reduced in this technique, making PEPE realizable in terms of cost and current technology. The cost of PEPE tends to be characteristic of modular and repetitive procurements. Not being dependent on future advanced, it involves little technological risk and is well-suited to MSI and LSI technology.

- Programmable -- The power and efficiency of the PEPE approach are available to the programmer without requiring the use of special skills or knowledge. The system designer, in determining whether functions are basically parallel or sequential, decides where functions will be executed and thereby what variables will be stored in PEPE. Then the programmer writes his programs straightforwardly in the Parallel FORTRAN (P-FOR) language. The ability to express vector operations and set specification in a natural way in this language means that the programs contain fewer statements, are easier to understand, and faster to debug. Furthermore, the program development for a PEPE-based system should be simpler and faster because PEPE, with its fully parallel/associative approach, provides a very general but efficient file structure built into the hardware.

PROTOTYPE BULK-FILTER DEVELOPMENT CONCEPT DEFINITION

The purpose of this concept definition study for the Advanced Ballistic Missile Defense Agency was to establish the feasibility of using associative-processing techniques for bulk filtering. During reentry, the natural breakup of an intercontinental ballistic missile (ICBM) booster tank produces several hundred objects which are visible to the radar. Treating all of these objects as potential threats in a tracking computer would immediately saturate the computational resources. There is a definite need for some form of preprocessing which can effectively and inexpensively recognize nonthreatening objects. This type of preprocessing is referred to as bulk filtering.

Several representative bulk-filtering problems, which illustrate the characteristics of the data base and the nature of the computational power desired, were examined in detail to determine the number and type of instructions that would be required to implement them.

Honeywell has developed an associative process or concept which fits very well to the bulk-filtering problem. This architecture is shown in Figure 11 and consists of:

- Dual-Control Units -- These units are operated simultaneously. One unit is assigned to input and correlation; the other controls arithmetic or processing operations.
- A Column of Identical PEs -- Each element possesses associative capability for input, memory for storing radar data, and 24-bit parallel arithmetic capability for the actual execution of bulk-filtering algorithms.

In this architecture, the number of radar "objects" per PE may be varied depending on the algorithm complexity, the threat load, and PE speed.

The exact tradeoff which yields the most cost-effective system for the tactical situation will depend on the future experimental results and on cost-effectiveness studies based on actual tactical scenarios.

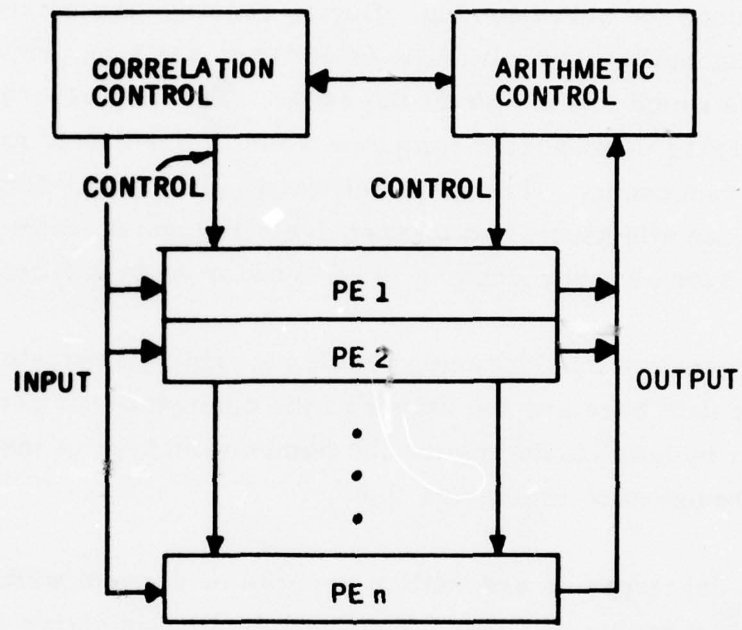


Figure 11. Bulk-Filter Associative Processor

A configuration was chosen for the future experimental system which provides one simple PE per track. This could provide the lowest cost and the desired flexibility, and further could be an excellent candidate approach for a deployable system. The flexibility required of the experimental system is provided through the use of a high-level language, P-FOR. This language is an extension of FORTRAN and includes parallel statements in addition to the standard FORTRAN language.

This study concluded that an associative/parallel approach was an ideal solution to the critical problem of bulk filtering and one which could be inexpensively tested and verified in an experimental system.

SECTION 3

DDP: PERSPECTIVE OF NEEDS BASED ON SITE DEFENSE EXPERIENCE

This section documents the independent perspective of a key former Site Defense software manager concerning the issues, payoffs, and research areas of DDP, based on Site Defense experience. The major areas covered in this section are DDP concepts and requirements, DDP payoffs, changes in Site defense requirements and their impact on the DDP Design Theory, Site Defense problems and their implications on DDP research, the design techniques and approaches used on Site Defense software development, and a summary of DDP research issues and recommendations.

DISTRIBUTED DATA PROCESSING OVERVIEW

Figure 12 is an introductory chart which asserts that distributed data processing provides the real-time control and decision for advanced ballistic missile defense systems. It is anticipated that all future BMD systems will have DDP; consequently, a DDP design technology and guidelines must be developed to meet the needs. A BMD system using distributed processing will typically consist of sensors, weapons, external interfaces, and a set of processing networks to provide a cohesive systems unit. The networks will consist of some combination of hardware, firmware, and/or software which will provide the throughput, memory, input/output which will be necessary to meet processing requirements of the system. The processing network implementations will be a function of the particular sensors and weapons and the particular external interface requirements that exist for a given system.

Those specific needs must be mapped into interconnected networks as shown in the center of Figure 12. This generic network consists of many nodes and connections. Specific implementations will be cost-effectively obtained based on two factors: the data-processing functional and performance requirements which are input to the design activity, and the DDP Technology Theory and guidelines which are currently under research.

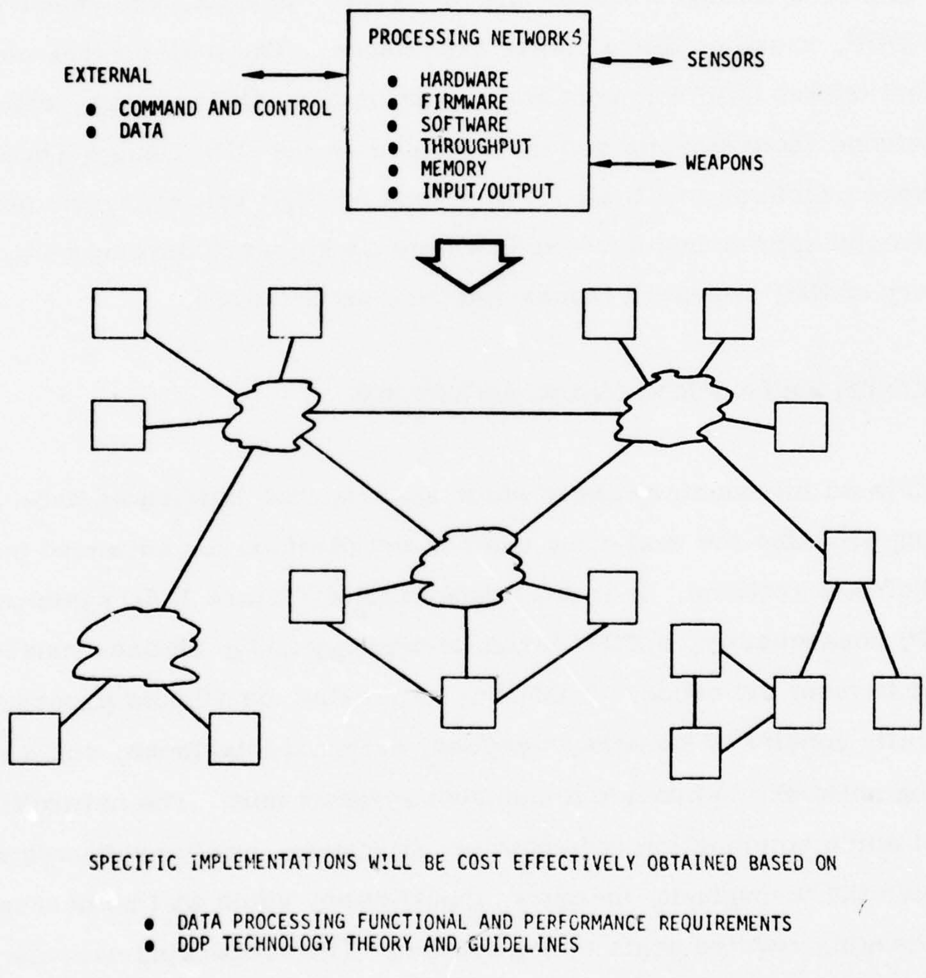


Figure 12. DDP Provides Real-Time Control and Decision for Advanced BMD Systems

THROUGHPUT REQUIREMENTS

Figure 13 estimates the projection of total force data processing system throughput requirements as a function of the date of the threat baseline and the two existing systems. Safeguard and Site Defense require approximately 100 and 1 billion instructions per second throughput. The future threat baseline will be strongly influenced by the proliferation and complexity of the threat. Both of these factors are monitored and modified by the treaties that have been taking place: if the treaties are successful then the lower end of the ban will be a more appropriate limit; if the treaties are unsuccessful, then the upper end would indicate the growth in throughput required. This would imply that a very significant increase in data processing throughput is needed. The total force structure will be growing by orders of magnitude implying that distributed data processing is really a vital technology. The alternative data processing solution of a very large mainframe presents problems which will be discussed subsequently. From Figure 13, it can be seen that the threat growth shows that flexibility and growth to very large throughputs is a necessary part of BMD data processing subsystems.

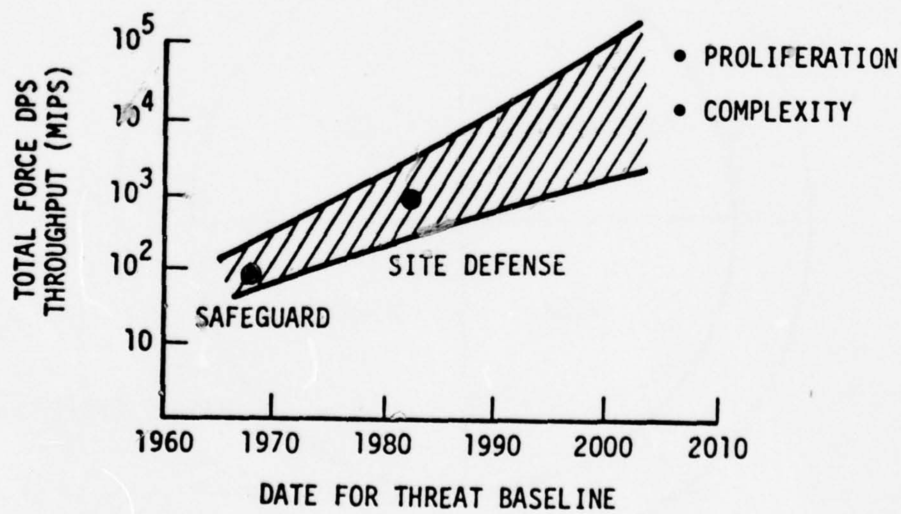
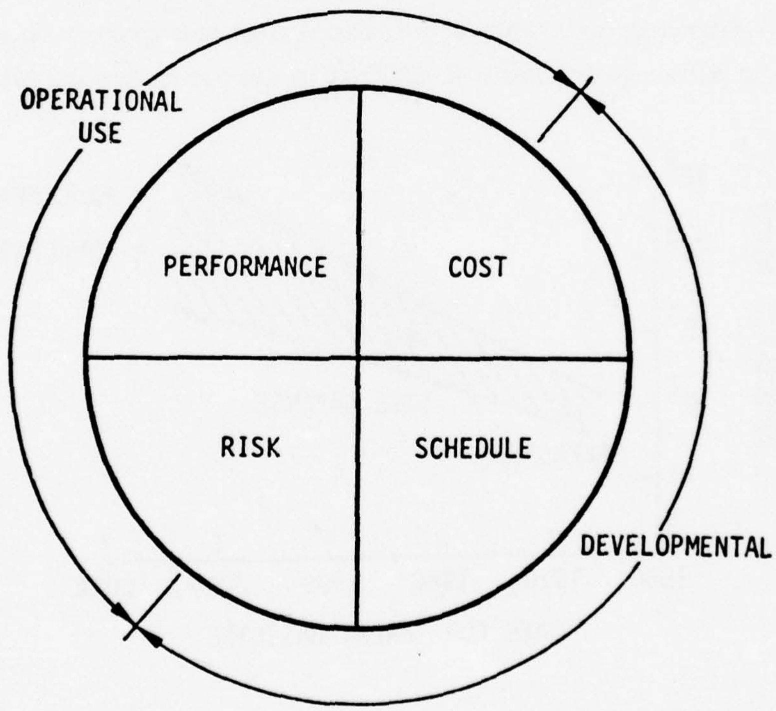


Figure 13. Projected Throughput Required Will Increase in Response to Threat

DDP PAYOFFS

Figure 14 indicates that DDP payoffs are realizable in all important aspects of BMD concepts. In any systems development program the progress and status of the system can be viewed in four major areas: performance, cost, schedule and risk. The four areas are continually traded off during development and also during the deployment. Figure 14 is divided into two major segments: operational use and development use. The developmental use is strongly influenced by the kind of schedule requirements which exist: the cost of the development, and the amount of risk which is taken in the development. Operationally speaking, performance is the key factor, but the cost and risk of operation are also important. Distributed Data Processing benefits all life-cycle phases, and the particular payoffs in each of these major areas are discussed in the following figures.



• ALL LIFE-CYCLE PHASES ARE BENEFITED

Figure 14. DDP Payoffs Are Realizable in all Important Aspects of Advanced BMD Concepts

PERFORMANCE: THROUGHPUT AND RESPONSE TIME

Figure 15 addresses two performance payoff attributes, throughput and response time, as depicted on typical curves. The uppermost curve, throughput, is one way of representing this constrained problem. A baseline Site Defense system threat scenario (module level) has been selected as a given, then depending upon the MIPS level that is required, one can read the curves to determine the most preferable architecture from a throughput standpoint. For example, the theoretical limit shown would indicate the minimum degree of concurrency required without taking into consideration data transfer and other overhead factors. As shown in the figure, less concurrency would imply that Architecture B would be the best configuration for fewer MIPS; as the MIPS requirement increases, it would be beneficial to change to Architecture A. These curves are discussed in the next several figures. The point is, that for a specific function or problem there are many ways to implement the solution, and depending upon the degree of concurrency and the architectures available (optimized or non-optimized), a "best" architecture could be determined from the standpoint of performance. The curves which are depicted in Figure 15 in a generic sense, require simulation and experimentation in order to quantify them. Particular architectures would have to be developed for the degree of concurrency plot shown. This is one of the recommended topics in the experiment plan (Volume VII). The second curve in Figure 15 shows response time. Here we have plotted a time delay (in milliseconds) against the degree of concurrency for three different parametric values of proximity of processors: zero, five, and ten nautical miles. One can observe that relatively longer timing delays due to queues, processing, and transport delays occur as the degree of concurrency is reduced and as the proximity decreases. (Here the degree of concurrency is interpreted to mean the degree of parallel processing that can take place within a single main frame such as the CDC7600.) As the degree of concurrency grows very large and processors have wider physical geometry separations, the reverse trend would be noted. To develop this particular curve as an analytical tool, simulation and experimentation as well as a specific system construct are required (see Volume VII, Experiments).

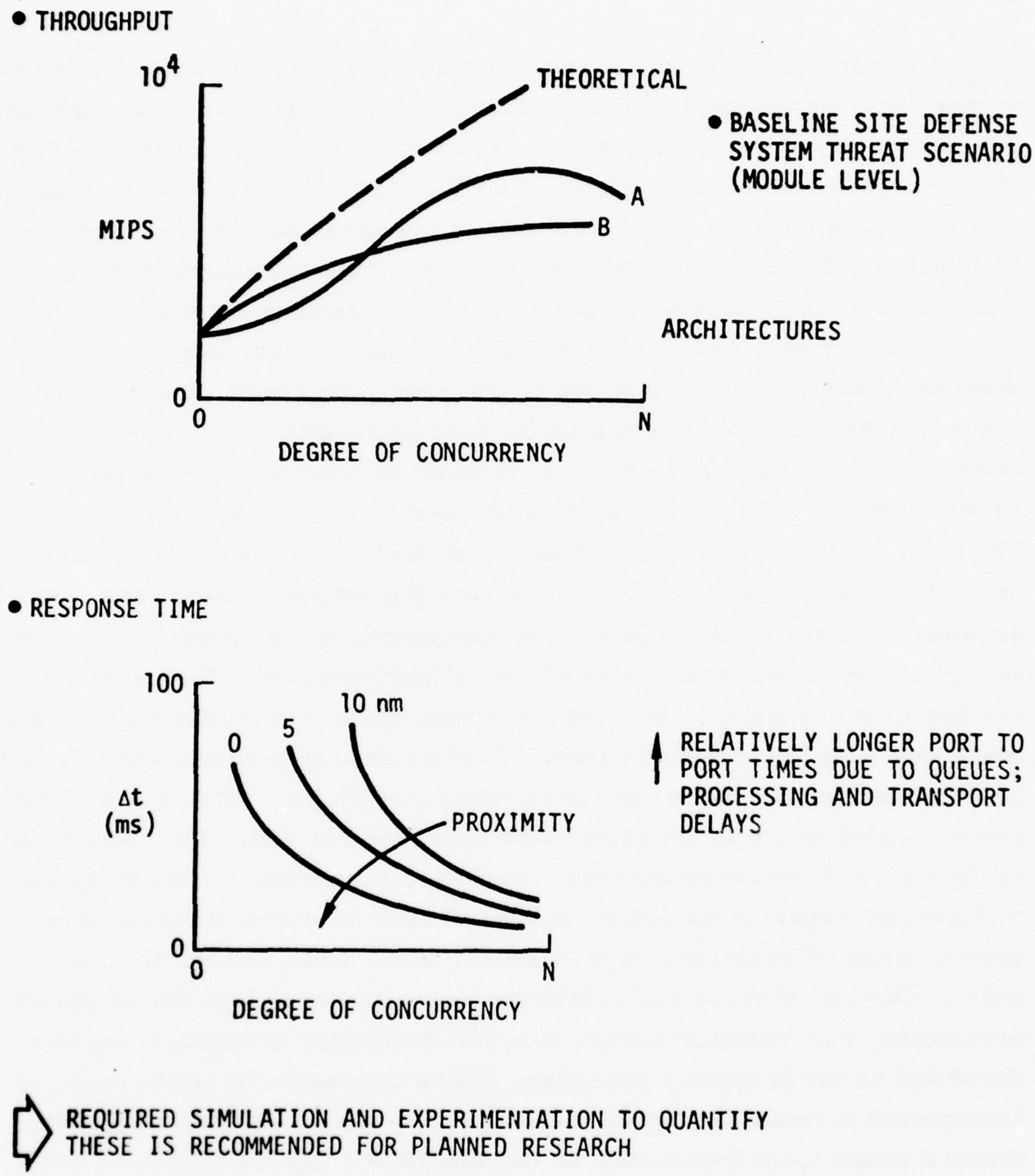


Figure 15. Payoff Attributes-Performance

ARCHITECTURE A: MULTICOMPUTER

In Figure 16 we discuss in more detail Architecture A from Figure 15, where Architecture A is here classified as a multicomputer. (Multicomputer defined as several stand-alone computers that are netted together.) The MIPS curve against the degree of concurrency curve is shown to have an increasing theoretically available value, but an application performance curve that actually reaches a peak at some specific level of concurrency. The peak value for the specific application is the optimum place to operate, for, if concurrency increases beyond this point, task splitting actually increases the data transfer required. As shown in the figure, increasing trends in throughput are due to use of multiple-independent registers which reduces scheduling conflicts (since the task could be spread out over more computers). Memory management is generally simplified. Since each computer would take care of its own memory management, memories might be smaller for remote or satellite locations. The central memory, if there was such a thing, would also be smaller because it would require only processing and storing pre-edited data from the remotes. So simplified memory management would result in effective increase in throughput, and would allow implementation of smaller independent tasks. In addition, a smaller definitive data base would result in less contention for resources, and there would be more firmware candidates that would allow some firmware implementation, offloading software tasks. On the other hand, certain features of Architecture A would tend to decrease the throughput: for example, it is possible to overload an individual task (since it is data driven). A single computer might not have enough capacity or throughput to handle all of the data that drives the particular task. Also there would be more data transfer interfaces to control, and as very large degrees of concurrency occur, the tasks might tend to get so small as to be unmanageable. The specifics of this curve are implementation dependent. It is necessary to develop a technique to map out these trends for different configurations and different architectures (see Volume VII).

INCREASING TRENDS IN THROUGHPUT--

- MULTIPLE INDEPENDENT REGISTERS REDUCE SCHEDULING CONFLICTS
- SIMPLIFIED MEMORY MANAGEMENT
- SMALLER INDEPENDENT TASKS
- DEFINITIVE DATA BASE--LESS CONTENTION
- MORE FIRMWARE CANDIDATES

DECREASING THROUGPUT TRENDS--

- POSSIBLE TO OVERLOAD INDIVIDUAL TASK (DATA DRIVEN)
- MORE DATA TRANSFER INTERFACES TO CONTROL
- TASKS GET SO SMALL AS TO BE UNMANAGEABLE

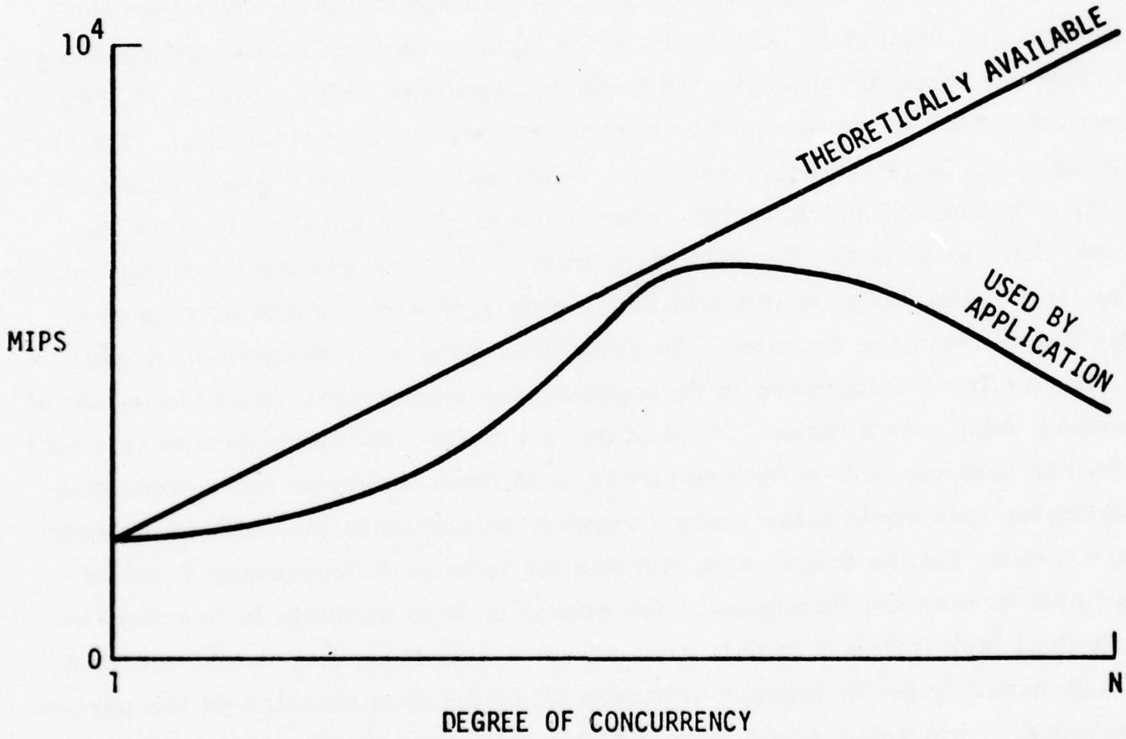


Figure 16. Architecture A: Multicomputer

ARCHITECTURE B: MULTIPROCESSOR

Figure 17 addresses Architecture B, a multiprocessor. [Multiprocessor here is defined to be where shared memory was used and servicing several Central Processor Units (CPUs)]. The theoretical limit goes up to some memory limit and cannot increase beyond that because the memory would constrain processing throughput. The application curve would tend to rise and the degree of concurrency would be somewhat below that of the memory limit. Increasing throughput trends would be achieved up to a point due to the availability of multiregisters which reduces scheduling contention. A large dynamic memory would be available which would increase throughput without a great deal of overhead assuming it was properly designed, and until its limit was reached. Interprocessor communications would be relatively simplified by the large memory; and single-processor failure could be accommodated. There is less risk of overdriving a task. Trends that would decrease the throughput would be: management of the memory becomes a large task in itself, and data base handling becomes more difficult and more time consuming (see also simulation experiment on this subject in Volume VII).

INCREASING THROUGHPUT TRENDS--

- MULTIPLE REGISTERS AVAILABLE REDUCES SCHEDULING CONTENTION
- LARGE DYNAMIC MEMORY AVAILABLE
- INTER-PROCESSOR COMMUNICATIONS SIMPLIFIED BY LARGE MEMORY
- SINGLE PROCESSOR FAILURE EASILY ACCOMMODATED
- LESS RISK OF OVERDRIVING A TASK BECAUSE OF LARGE MEMORY

DECREASING THROUGHPUT TRENDS--

- MEMORY MANAGEMENT BECOMES A LARGE TASK
- DATA BASE HANDLING MORE DIFFICULT

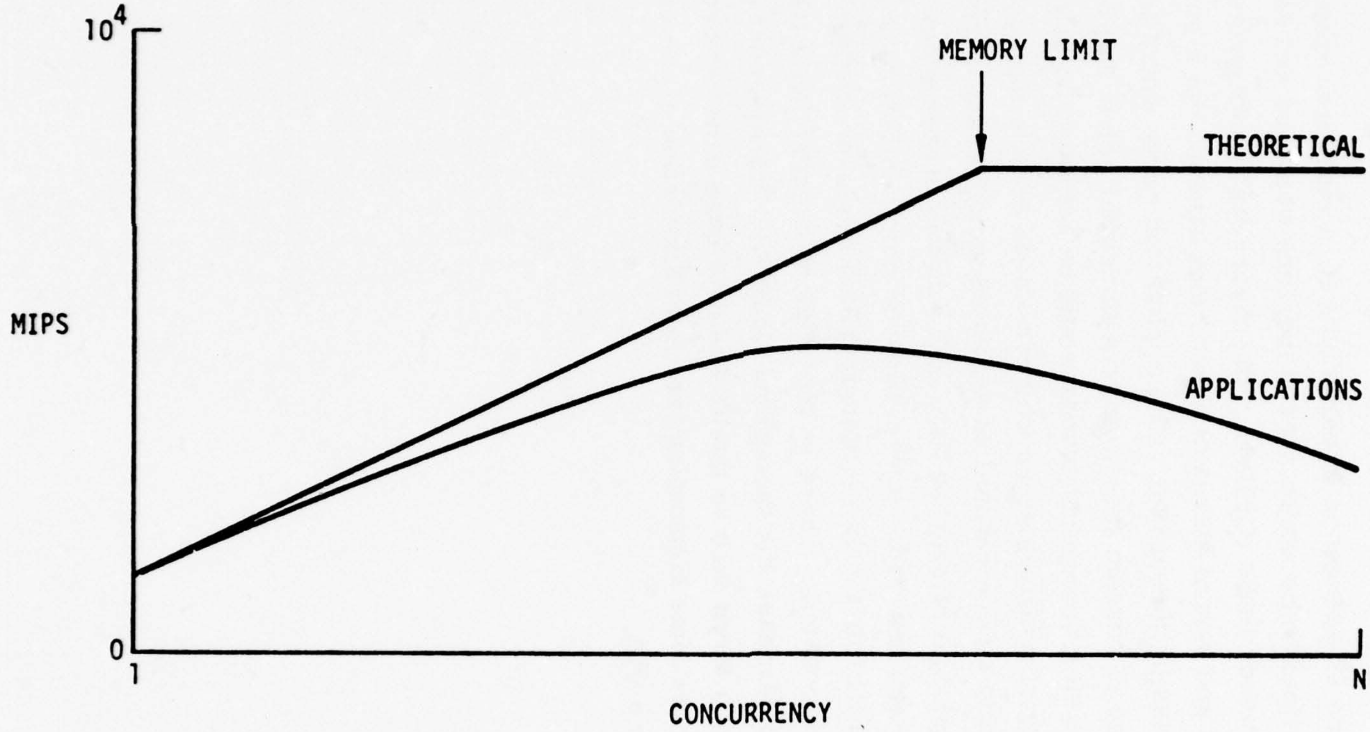


Figure 17. Architecture B: Multiprocessor

SYSTEM INTEGRITY

Table 3 continues the discussion of performance attributes under the subheading of System Integrity. The first is survivability. The survivability equation would be a function of the node's design and how the failure of a given node would affect the performance of the whole system. In general, however, it is possible to postulate a function that is determined by the probability of kill at a V-node, or vulnerability node, raised to a power. The implied design attribute would be to configure many nodes sufficiently separated so that survivability of one would not damage another. Since the survivability issue tends to be a system problem rather than a data-processing problem, it should be worked at the system level. There is a direct interface here between the system designers and the data-processing designers, where the notion of many nodes would be mapped into processors of a given size, a given capability and connections that would provide sets of data processing nets. Failsoft behavior (in terms of redundancy of units and semi-autonomous nodes) implies that there would be a design requirement of no weak links: if a failure did occur, there would either be a redundant component to take over its function, or it would be capable of segmenting into smaller units that were semi-autonomous and could continue working. Reliability, availability and maintainability would include fault-tolerance characteristics and dynamic restructuring characteristics. Fault tolerance would be based upon switching methods and error detection/correction methods which would produce a design attribute of a virtual repair, so that there would be little or no impact or observable performance degradation. The goal would be that the fault would be transparent to the application programming and the system performance. Dynamic restructuring would lead to configuration flexibility so that we could restructure and reconfigure the system. As certain elements failed we could reallocate the resources and reroute the connections such that a major capability still remained. The final bullet in the table shows some topics on graceful saturation. The DP subsystem should be capable of a virtual capacity so that as

TABLE 3. PAYOFF ATTRIBUTES - PERFORMANCE

	IMPLIED DESIGN ATTRIBUTE
• SYSTEM INTEGRITY	
• SURVIVABILITY	
$P_S \approx \sum (1 - P_{K_{V\text{-}NODE_i}})^n$	MANY NODES
• FAIL SOFT BEHAVIOR	
- REDUNDANCY	NO WEAK LINK
- SEMI-AUTONOMOUS MODES	
• RELIABILITY/MAINTAINABILITY/AVAILABILITY	
- FAULT TOLERANCE	VIRTUAL REPAIR
- SWITCHING	
- ERROR DETECTION/CORRECTION	
- DYNAMIC RESTRUCTURING	CONFIGURATION FLEXIBILITY
- RESOURCE ALLOCATION	
- CONNECTIVITY REROUTING	
• GRACEFUL SATURATION	VIRTUAL CAPACITY
- LOAD SHEDDING	
- DYNAMIC RESOURCE MANAGEMENT	

the number of objects becomes very large, or the arrival density becomes very high, the subsystem has load-shedding capability for reducing those objects that are not desirable, and dynamic resource management so that fewer resources can be allocated per object. This virtual capacity is a design attribute that could be implemented in the software or the firmware or the hardware.

COST

This subsection on cost attributes is addressed in Table 4. Here we've simply listed all of the attributes that have a cost payoff. (These major attributes could be explained at some length but this list is fairly self-explanatory.) The DDP cost payoff attributes will reduce total life-cycle cost in all of the areas named on the chart. For example, the dynamic restructuring/fault-tolerance error detection/correction category etc., improves reliability and availability of the system. This, in turn, makes it possible to trade off the cost of implementing that attribute versus the cost of having lower reliability/availability. In general the design can be set for a cost payoff and improved R and A. Modularity is going to improve the mean time to repair and it will enhance the growth capability. If each of the elements is modular in nature it is possible to make changes or increase the number of processors in the net go to more complex systems in a more cost-effective/efficient manner. Finally, another cost benefit would be the testability. As you distribute the system, distribute the data processing, the testability can be partitioned more cleanly, in general, and the interface messages which have to go between the units can be catalogued and monitored. One could argue that this same structure would be used in large main frames and this is true; however, with distributed data processing the testable modules would tend to be smaller in size which would make the problem somewhat simpler because there would be less contention for memory and other resources at a local CPU.

TABLE 4. PAYOFF ATTRIBUTES - COST

- DDP COST PAYOFF ATTRIBUTES REDUCE TOTAL LIFE CYCLE COST

ATTRIBUTE	COST PAYOFF
ADAPTABILITY/FLEXIBILITY/GROWTH	REVISION EASIER AND MINIMIZES BREAKAGE
ACCESSIBILITY	REVISION, OPERATIONS/SUPPORT SIMPLIFIED
COMPATIBILITY	INTERFACES AND REVISIONS ENHANCED
THROUGHPUT	SIMPLIFIES SOFTWARE IF SUFFICIENT MARGIN
DEPLOYABILITY	SIMPLIFICATION
DYNAMIC RESTRUCTURING	IMPROVES RELIABILITY/AVAILABILITY
FAULT TOLERANCE	IMPROVES RELIABILITY/AVAILABILITY
ERROR DETECTION/CORRECTION	IMPROVES RELIABILITY/AVAILABILITY
REDUNDANCY	IMPROVES RELIABILITY/AVAILABILITY
GRACEFUL SATURATION	REDUCES BRITTLENESS
HARDNESS	REDUCES VULNERABILITY, HENCE NUMBERS
HARDWARE SIMPLICITY	DEVELOPMENT, OPERATION & SUPPORT SIMPLIFIES
MAINTAINABILITY	REDUCES OPERATION & SUPPORT COST, IMPROVES AVAILABILITY
MODULARITY	IMPROVES MTTR, ENHANCES GROWTH
STANDARDIZATION	FACILITATES DEVELOPMENT
TESTABILITY	FACILITATES DEVELOPMENT

RISK

Table 5 addresses the payoff attributes of risk. Distributed data processing does have the potential to reduce the risk that is associated with the performance, cost, and schedule. (Here risk is defined to be basically an overlay to the other three performance payoff attributes and this overlay would be examined in detail for each of those three areas.) Development risk may be reduced by partitioning the development into tractable subsets. This would tend to simplify the definition of requirements and design, and by defining requirements and design in a simpler way we are able to construct cleaner modules, cleaner interfaces, and simpler ones which would be more easily managed; consequently the risk would tend to be reduced. This partitioning would also optimize and control configuration of interfaces, reduce the volume of the central data base (due to the statement made earlier that it has to handle only the edited data rather than large volumes of raw data). There is enhanced visibility of progress because each of the particular units or elements can be generated and certified step by step. Partitioning would also provide inherently separable and testable packages, enhance deployability, provide adaptability for revision, and provide corresponding cost improvements due to all of the reasons cited above. Operational risk is reduced by providing greater assurance of meeting planned and projected performance objectives. It is possible to increase the survivability of the system, add operational flexibility and reduce brittleness, reduce sensitivity to local processing failures, add concurrency for better throughput and availability, and enhance graceful degradation. In summary the payoff attributes in the risk area map into all of the other areas. In general there is better visibility, and higher confidence that the product which will be produced will operate properly.

TABLE 5. PAYOFF ATTRIBUTES - RISK

- DISTRIBUTED DATA PROCESSING HAS POTENTIAL TO REDUCE RISK ASSOCIATED WITH PERFORMANCE, COST AND SCHEDULE

- DEVELOPMENT RISK IS REDUCED BY PARTITIONING THE DEVELOPMENT INTO TRACTABLE SUBSETS
    - SIMPLIFIES DEFINITION OF REQUIREMENTS AND DESIGN
    - OPTIMIZES AND CONTROLS CONFIGURATION INTERFACES
    - REDUCES VOLUME OF "CENTRAL" DATA BASE
    - PROVIDES ENHANCED VISIBILITY OF PROGRESS
    - PROVIDES INHERENTLY SEPARABLE, TESTABLE PACKAGES
    - ENHANCES DEPLOYABILITY
    - PROVIDES ADAPTABILITY FOR REVISION
    - PROVIDES CORRESPONDING COST IMPROVEMENTS

- OPERATIONAL RISK IS REDUCED BY PROVIDING GREATER ASSURANCE OF MEETING PLANNED AND PROJECTED PERFORMANCE OBJECTIVES
    - INCREASES SURVIVABILITY
    - ADDS OPERATIONAL FLEXIBILITY; REDUCES BRITTLENESS
    - REDUCES SENSITIVITY TO LOCAL PROCESSING FAILURES
    - ADDS CONCURRENCY FOR BETTER THROUGHPUT AND AVAILABILITY
    - ENHANCES GRACEFUL DEGRADATION

REAL-WORLD DEVELOPMENT REQUIREMENTS

Table 6 is important in that it shows real-world conditions based upon situations experienced in Safeguard and Site Defense, and observations that were made on those two systems. The DDP design theory must consider these real-world conditions as they are manifested in the data-processing requirements. Throughout the development of BMD systems the threat is continually changing due to intelligence data, treaties, or other factors that would cause the threat to be modified. The threat changes typically would be in terms of proliferation in numbers, revision in the characteristics, penetration aids or offensive tactics. Any of these changing threat characteristics would tend to cause major impact on software or data-processing design if they occurred midway in the development. The characteristics revisions referred to include items like the RCS or the observables data, discrimination characteristics, and other data that is used to track or detect or identify what the target might be.

The DDP design technology must allow for these changes in threat without major impact and restructuring or redesign of the entire data processing network every time the threat changes. Next is changing mission. In general, there will be revisions of defended areas and points, the required defense levels may have to go up or down, command and control variations will occur, and security-level changes will occur. The defended areas and points would be determined by National Command Authority (NCA). Each of these changing missions would have broad impacts on the DDP structure and consequently the design technology should allow for it. Changing and understanding phenomenology, nuclear effects, target signatures, bulk filtering are three very important areas which have had major impact on the Site Defense program. Understanding the models for these effects, mapping those into requirements on software that can be implemented in real time, responding to changes as the models change or as additional data comes in that affect the models, are important DDP design technology attributes.

TABLE 6. CHANGING ELEMENTS WHICH MUST BE CONSIDERED IN DESIGN THEORY

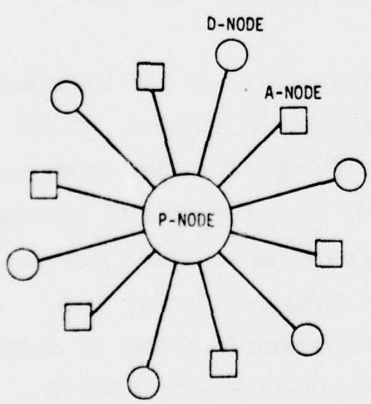
- CHANGING THREAT (INTELLIGENCE DATA)
  - PROLIFERATION IN NUMBERS
  - CHARACTERISTICS REVISIONS
  - PENETRATION AIDS
  - OFFENSIVE TACTICS
- CHANGING MISSIONS
  - DEFENDED AREAS/POINTS
  - DEFENSE LEVELS
  - COMMAND AND CONTROL VARIATIONS
  - SECURITY LEVELS
- CHANGING/UNDERSTANDING PHENOMENOLOGY
  - NUCLEAR EFFECTS
  - TARGET SIGNATURES
  - BULK FILTERING
- EVOLVING SUBSYSTEMS
  - SENSORS
  - WEAPONS
  - PROCESSING
  - COMMUNICATIONS/DATA TRANSFER
- EVOLVING TECHNOLOGY IN DATA PROCESSING
  - PROCESSING DEVICES
  - MEMORY MEDIA
  - DATA TRANSFER
  - SOFTWARE DESIGN AND DEVELOPMENT

The fourth major category is evolving subsystems. In most DOD procurements the subsystems of a major system are developed somewhat in parallel. This includes the sensors, the weapons, processing, and communications and data transfer. As the subsystems evolve there must be give and take and configuration control among them. Many systems, for example, would develop all of these items and then eliminate any differences or inconsistencies, missed performance requirements, or failure of the hardware to do its job, through software modifications. However, this can lead to very costly changes in software configurations and many cost overruns in projects throughout the nation and also through all the different branches of the services. One project even took an opposite approach which was to build software first and constrain the hardware to match. In any case, the evolving subsystems will have a dramatic impact on the interfacing of the software and the hardware in the system. Software must control and operate that hardware, and processing network must be capable of responding to evolutionary changes in those items. Further, a particular sensor may be completely replaced with a newer variety. The same thing may be true of a weapon, and as these changes occur the impact on the processing should be manageable. If the processing is designed to be totally integral so that a change in a weapon would impact the entire set of tasks in a software package then there would be a major problem. Consequently, the DDP design technology should attempt to optimize these types of changes.

The last category involves technology in data processing. It has been said that the half life of data processing technology is about 6 months, which makes it very difficult to always have the "latest thing" in major systems. The procurement lead time is considerably longer than the rate of change of the technology, so consideration should be given to processing devices, memory media, data transfer, and techniques in software design and development that may impact and even obviate the technology that is used on a given system. If one could determine a convenient way to always upgrade to the newer technology with a faster response time in a cost-effective manner, this would be a noteworthy accomplishment for the DDP design theory.

CENTRALIZED DP

Centralized and distributed systems are compared in Figure 18. First is an example of a centralized system which can be depicted as a star network characterized as three types of nodes. There's a P-node, which is the processor, a D-node for data and an A-node for action node. Our assumptions are that all data is gathered at data nodes or D-nodes, all data flows back to the central processor or P-node, and then all action flows out to an action node. All control resides at the central processor. To function successfully the central processor must be operational, the line capacity and timing delays must be sufficient to handle the loads, and contention problems at the CP must be manageable. As the number of D-nodes and A-nodes increases, of course, the contention problem gets more severe. Types of problems that occur with this system design are that the central processing becomes a huge message switcher; it simply receives messages and sends messages and switches them. The function scheduling is extremely important. There is also binary vulnerability of the system; if you take out the P-node the system is down, unless certain types of redundancy are built-in (which would tend to decentralize it).



ASSUMPTIONS:

(1) ALL DATA GATHERED AT NODES (D-NODE)
(2) ALL DATA FLOWS BACK TO CENTRAL PROCESSOR (CP) OR P-NODE
(3) ALL ACTION FLOWS OUT TO ACTION NODE (A-NODE)
(4) ALL CONTROL RESIDES AT CP

NECESSARY CONDITIONS TO FUNCTION SUCCESSFULLY:

(1) CP MUST BE OPERATIONAL
(2) LINE CAPACITY MUST BE HIGH ENOUGH TO HANDLE LOADS
(3) CONTENTION PROBLEMS AT CP MUST BE MANAGEABLE

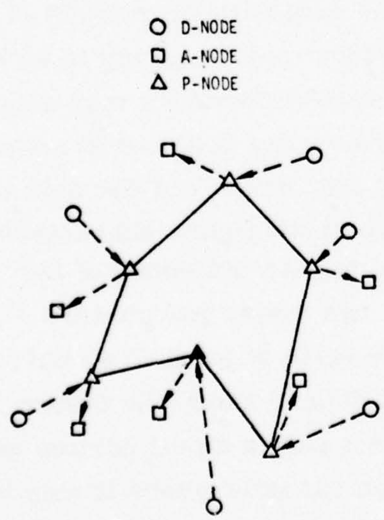
PROLEMS:

(1) CP BECOMES A HUGE MESSAGE SWITCHER
(2) FUNCTION SCHEDULING BECOMES EXTREMELY IMPORTANT
(3) BINARY VULNERABILITY OF SYSTEM

Figure 18. Centralized System

SIMPLE NETWORK DP

Figure 19 represents a simple network system. Here again there are the three nodes; data node, action node, and processor node; however, in this case we have multiple processor nodes. Features include the communications system required and the control system required but these may be separable items. Whether the data action processor unit is capable of functioning alone or not must be resolved. The single P-node is designated a commander as shown by the solid triangle in the figure. The control problem of this type of network is a contemporary topic and the manner in which the control is implemented has been under considerable study. The benefits of this type of system are that communications are simplified, in that messages may be routed to the appropriate place. Growth potential is maximized so that it is possible to add on, vulnerability is minimized since taking out one or more of the nodes could be counteracted by another node, the control of action and data nodes is simplified, and party-line versus direct communications can be implemented, and the tradeoff can be done for the best solution. One problem that may arise is that of resource contention and resource management and how to really establish where the major control shall be. (See also Volume II and Volume VII).



FEATURES:

- COMMUNICATIONS SYSTEM REQUIRED } SEPARABLE
- CONTROL SYSTEM REQUIRED } SEPARABLE
- DAP UNIT CAPABLE OF FUNCTIONING ALONE?
- SINGLE P-NODE DESIGNATED COMMANDER ▲

BENEFITS:

- COMMUNICATIONS SIMPLIFIED
- GROWTH POTENTIAL MAXIMIZED
- VULNERABILITY MINIMIZED
- A-NODE + D-NODE CONTROL SIMPLIFIED
- PARTY-LINE vs DIRECTED COMMUNICATIONS
-

PROBLEMS:

- RESOURCE CONTENTION/MANAGEMENT?

Figure 19. Simple Network System

DISTRIBUTED DATA BASE

Some of the payoffs of distributing the data base (Figure 20) include the fact that local data remains local, i.e. that information which is needed to control a sensor, for example, need go no further than the processor dedicated to controlling the sensor. Global data can be controlled at a node and then sent to users so that it provides a function of coordination and monitoring and integration of all of the information; then correct action orders or constraints would be sent to all the users. The hierarchical design for control and efficiency may be a payoff although this is an issue of some discussion and whether it is indeed a payoff or a problem area is something to be resolved. Distributing the data base will enhance graceful degradation since portions of the data base being eliminated would not affect the system in any way relative to that of a single central data base. It does allow the transfer of processed data and consequently reduces the bandwidth. (In other words the raw data is processed and edited at satellite locations and then sent to its appropriate destinations.)

Issues concerning distributing the data base would be the integrity and consistency of the data base, updating procedures, stability of the data base, staleness of global data, data transfer rates, the impact on the algorithm design, message design, and sizing. The issues deal with the subject of integrity in the data base, since several processors may be trying to write into the data base at the same time and the physical differences may affect those updates. There will be timing delays. If stability goes out of control, reading and writing of the memory, and reading and writing of the data will become an issue. The scale shown at the bottom of the figure characterizes some of these issues and payoffs. It is possible to map staleness of the data base as a function of data transfer rate for two major parameters - the number of processors in the network and the ratio of local to global data base. As seen in the figure the more local data in hand, the faster the response time. This is intended to imply that only a small portion of data (a lower volume of data) is required at a central area where it may be

PAYOFFS	ISSUES
LOCAL DATA REMAINS LOCAL	INTEGRITY/CONSISTENCY
GLOBAL DATA CONTROLLED AT A NODE AND SENT TO USERS	UPDATE PROCEDURES
HIERARCHICAL DESIGN FOR CONTROL AND EFFICIENCY	STABILITY
GRACEFUL DEGRADATION ENHANCED	STALENESS OF GLOBAL DATA
ALLOWS TRANSFER OF PROCESSED RAW DATA (REDUCES BANDWIDTH REQUIRED)	DATA TRANSFER RATES
	ALGORITHM DESIGN IMPACT
	MESSAGE DESIGN
	SIZING

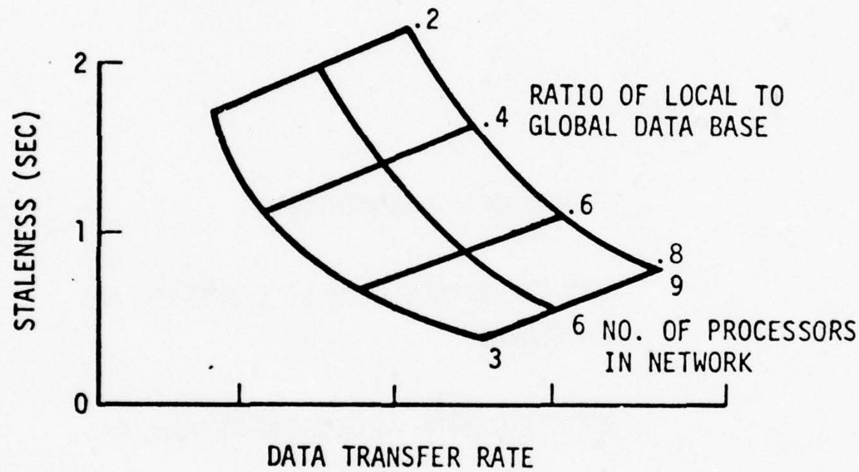


Figure 20. Distributed Data Base

classified as stale. In general the more processors in the network, the more staleness that will occur. To calculate accurate values for this figure would require simulation and specific assumptions concerning numbers of processors, physical distance, and magnitudes of the data base. Simple hand calculations were used for this figure, and the assumptions that were used produced these figures. For specific implementation, however, additional research would be needed and the design guidelines that would go along with the distributed data base would have to be resolved.

RESEARCH ISSUES OVERVIEW

DDP research issues (Table 7), including origins of payoff attribute categories, are discussed in this subsection. The origins include evolving BMD requirements, BMD experience from Site Defense and Safeguard, and evolving data-processing technology. Each of these three areas would, in fact, generate research questions that would give way to major research topics in the next phase of effort. The four major payoff attribute categories each give rise to specific research issues, as discussed on the following pages.

TABLE 7. RESEARCH ISSUES

- ORIGINS
  - EVOLVING BMD REQUIREMENTS
  - BMD EXPERIENCE FROM SITE DEFENSE AND SAFEGUARD
  - EVOLVING DATA PROCESSING TECHNOLOGY
- PAYOFF ATTRIBUTE CATEGORIES
  - PERFORMANCE (THROUGHPUT, INTEGRITY, RESPONSE, ETC.)
  - COST
  - SCHEDULE
  - RISK (PERFORMANCE, COST, SCHEDULE)

SITE DEFENSE REQUIREMENTS CHANGES

Table 8 lists the requirements changes that have been observed and their impact on the Site Defense System. A major change occurred when the tank breakup model had a 25 dB increase at the very beginning of the Site Defense program; the impact was a major redesign of all precommit functions. Indicated DDP research is a special function processor architecture and a flexible structure. For example, as the change occurred it was so large as to completely obviate the whole design of precommit; it became important to restructure all of the requirements at the system level. The impact of these on the data-processing level was major: it resulted in higher loads, faster response times and higher data rates, and higher track rates on the objects. It also resulted in more complex algorithms. The indicates research would be a special processor for some of these areas, perhaps in terms of firmware, special component, or a chip that could perform these functions very rapidly. As the problems became modified, the chip, for example, could be replaced by a new one. This flexible structure would be an optimum way to design a system. The system solution depends so heavily on the data-processing solution that it must be an integral part of the design solution.

The RV wake model was the second area of change. This change in the wake model caused an impact on the radar dynamic range, which caused a redesign in precommit and radar control algorithms. The indicated DDP research is application and control of special-purpose processing. For example, as these algorithms became more complex and more difficult to design, the processing loads that went along with them were also more difficult. The implementation for timing and threshold control and loading were much more difficult than prior to this problem. Indicated research would be to develop a way to apply and control that special-purpose processing, to insert it into the design structure where it did not previously exist, or where it existed in a much simpler way, without impacting the major design of the tasks and the hardware of the data processing system.

TABLE 8. REQUIREMENTS CHANGES

THE IMPACT ON THE SITE DEFENSE SYSTEM SUGGESTS NEEDED RESEARCH AREAS IN DISTRIBUTED DATA PROCESSING

CHANGE	IMPACT	INDICATED DDP RESEARCH
TANK BREAKUP MODEL 25 dB INCREASE	MAJOR REDESIGN OF PRECOMMIT FUNCTIONS	• SPECIAL FUNCTION PROCESSOR ARCHITECTURE • FLEXIBLE STRUCTURE
RV WAKE MODEL	RADAR DYNAMIC RANGE PROBLEM, REDESIGN PRECOMMIT AND RADAR CONTROL	• APPLICATION AND CONTROL OF SPECIAL PURPOSE PROCESSING
DEFERRAL OF SPRINT MISSILE	RETUNING OF PROCESS DATA BASE; UNNECESSARY SOFTWARE STRUCTURE	• EFFICIENT WAY TO REOPTIMIZE STRUCTURE AND DATA BASE
EMPHASIS ON KMR RATHER THAN TACTICAL	MANY CHANGES TO DATA BASE; UNNECESSARY SOFTWARE STRUCTURE	• EFFICIENT WAY TO REOPTIMIZE STRUCTURE AND DATA BASE
DEGRADED OPERATION REQUIREMENT DELETED	COST EFFECTIVE TO ADD MODULES	• COST EFFECTIVE WAY TO IMPLEMENT DEGRADED OPERATION
THREAT HARDNESS	POSTCOMMIT DESIGN AND DATA BASE	• PARTITIONING THEORY FOR SPECIAL PURPOSE PROCESSING
RADAR OPERATIONAL REQUIREMENTS	RADAR SCHEDULING CONSTRAINTS AND COMPLEXITY	• CONTROL AND SCHEDULING TECHNIQUES

The third area of change was deferral of the Sprint missile. The impact of that was returning of the process data base. It also resulted in an unnecessary software structure, since all of the missile software tasks tended to drive the design to a certain extent. There needs to be an efficient way to reoptimize the structure and data base without major cost and schedule impact. These reoptimizations are important in order to continually maintain a reasonably active, reasonably optimum data-processing performance. (The major reason for deferral was that the missile was deferred out of the validation program and was later intended to be put back in. The replacement missile, however, was different from the original Sprint missile, consequently all of the early work that was done on Sprint had to be updated to correspond to the newer weapon. This gets into the area described in an earlier figure about evolving subsystems.) As weapons evolve, major portions of the data-processing design have to be updated and those major updates can have dramatic impact on the cost, schedule, and performance of the system. Thus, the message for DDP research is to devise a scheme for efficient reoptimization of the structure and data base of the processing.

The fourth major change was that emphasis was placed on KMR rather than tactical operations. This action resulted in major changes to the data base and also unnecessary software structure, since only a subset was taken to KMR for testing purposes. Here again a development advantage would have been a more efficient way to reoptimize the structure in the data base rather than taking a straight subset as it existed. (It should be noted, however, that the Site Defense system did have a good design, in that reinitialization of the data base using only the subset of the structure that was required for KMR was adapted in a reasonably straightforward manner.)

Next, degraded operation requirements were deleted. It was found that, at the system level, it was more cost effective simply to add another defense module, consisting of multiple computers and missiles, etc., rather than to implement the software, operating system, test apparatus, etc. with degraded operation capability. The conclusion reached today may, of course,

be significantly different from that reached six or seven years ago owing to the new data processing technology. Today, the degraded operation might be done in a more straightforward manner. The indicated research is a cost-effective way to implement degraded operation. There is significant work being done and many applications are already in use for monitoring with microprocessor: determining what portions of the system are degraded; developing alternate tactics and operating rules that could be implemented in the software or in the digital processor. The research should pave the way for a cost-effective implementation of degraded operation. (Degraded operation can be defined as operating when certain elements of the system are unavailable or degraded due to battle damage.)

Threat hardness had a major impact on the post-commit design data base and the indicated research is a partitioning theory for special-purpose processing. Threat hardness will affect the intercept miss distance, placement of intercept points, probability of kill, and hence the number of missiles or weapons that are allocated to a given target (whether a salvo or a shoot-look-shoot type tactic is used); these major effects are strongly dependent on the threat hardness, which in turn is strongly dependent on the intelligence data we obtain for purposes of design requirements. The partitioning theory for special-purpose processing would account for various levels of changes such as threat hardness or other modifications to the threat that occur during the development of the system. It should be possible to break out or partition those portions of the processing that are driven by this type of change.

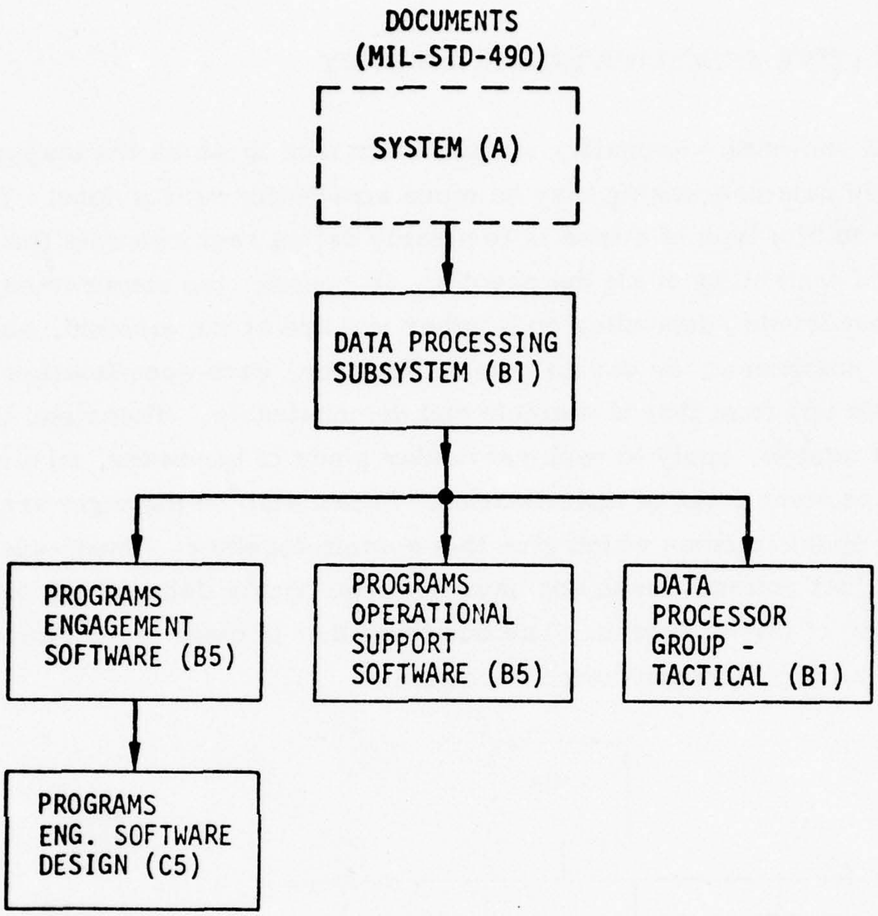
In discussing radar operational requirements, major impact was due to the radar scheduling constraints and complexity. At the beginning of the program the radar scheduling task was fairly straightforward, at least with respect to where it is now at the end (or the current stage) of the project. As the radar began to evolve, more and more was understood of how the radar would operate and how the cost factors would modify the radar performance. For example, there were continual design-to-cost tradeoffs on the radar as it

evolved, and certain decisions which were made in the radar implementation which had impact on the software. More and more constraints were added which gave rise to more and more complex algorithms and radar scheduling. Additionally, some of the other factors mentioned in the list such as wake models or tank breakup models, tended to give rise to different types of waveforms than were used initially. The indicated research would be control and scheduling techniques so that the implementation of these many changing constraints could be implemented in a more cost-effective manner. If it were possible to design a distributed concept with scheduling techniques as a modular component so that as these things developed and changed they would be more efficient, this would be a desirable factor (see Volume 4). However, the tradeoff between development cost and operational use must be considered. The impact of radar operation requirements during development was to cause a lot of rework and a lot of changes in radar scheduling algorithms. Operationally, the impact would be to reduce the amount of pulse repetition frequency (PRF) one could obtain with the radar, and reduce the duty cycle that could be obtained due to the complexity. So both of these problem areas appear as areas of research.

SPECIFICATION HIERARCHY

Figure 21 illustrates a MIL-STD-490 specification tree. At the top level there is a system spec (an (A) spec) which then feeds into the requirements on the data-processing subsystem or B1 specifications which further maps into engagement software specs, the B5, programs operational support software, and data processor group tactical. The engagement software (B5) spec then maps into the engagement software design, which is the C5 spec. In the current 490 protocol, these specifications are the important ones for controlling the requirements in the development of the hardware and software in the data-processing subsystem. Payoffs that would be mapped into the various specifications are listed at the bottom of the figure; the data indicate where in these various specifications particular payoff would be implemented. The important point is that payoffs, as they are currently categorized, tend to map into several different specifications. For example, effectiveness is really given at the A spec level but it is implemented down in the C5 spec level; whether or not the A spec requirements are met depends on how well the C5 spec is done.

Another factor is survivability, which is given the A spec, and also data processor group tactical where hardness requirements, etc., are mapped. The cost at the bottom of the payoff's list is really impacted by all the specifications although none of them really address it directly. Cost is only an implicit function in terms of the specifications. Third from the bottom is throughput, in which the B1 level a high throughput requirement is given, then down at the C5 level where program's engagement software design specs are given, the equations and the logic that is actually implemented will control how much throughput occurs under a given load condition. Not shown on the figure are a whole host of interface specifications which serve to glue together all of the various specification documents. These interface specs are very important in terms of pinning down and explicitly defining the data-processing requirements.



PAYOFFS	A SYS	B1 DPS	B1 DPGT	B5 PES	B5 POSS	C5 PESD
EFFECTIVENESS	●					●
RELIABILITY	●	●	●			
SURVIVABILITY	●		●			
MAINTAINABILITY	●					
AVAILABILITY	●	●				
GROWTH	●					●
DEPLOYABILITY	●		●			●
ROBUSTNESS	●		●			●
PREDICTABILITY						●
ADAPTABILITY (R.T)				●		●
ADAPTABILITY (DESIGN)				●		●
GRACEFUL SATURATION				●		●
THROUGHPUT		●				●
RESPONSE				●		●
COST	●	●	●	●	●	●

Figure 21. Specifications - Payoffs

ALTERNATIVE SPECIFICATION HIERARCHY

Figure 22 shows an alternative specification tree in which the mapping of distributed data processing may be more straightforwardly done. The objective in this type of a tree is to clearly define requirements for accountability and testability of all the payoffs. Of course, the requirements are at different levels, depending on whether you are at the element, sub-element, component, or device level. However, each specification should produce an end item that is testable and demonstrable. These end items would, of course, apply to each particular piece of hardware, firmware or software at some level of specification. Shown also on the right are the interface specifications which glue this system together. These specifications are just notional ideas and have not been firmly detailed out to test the validity of this approach. The one thing that is clear is that accountability is an objective that must be met.

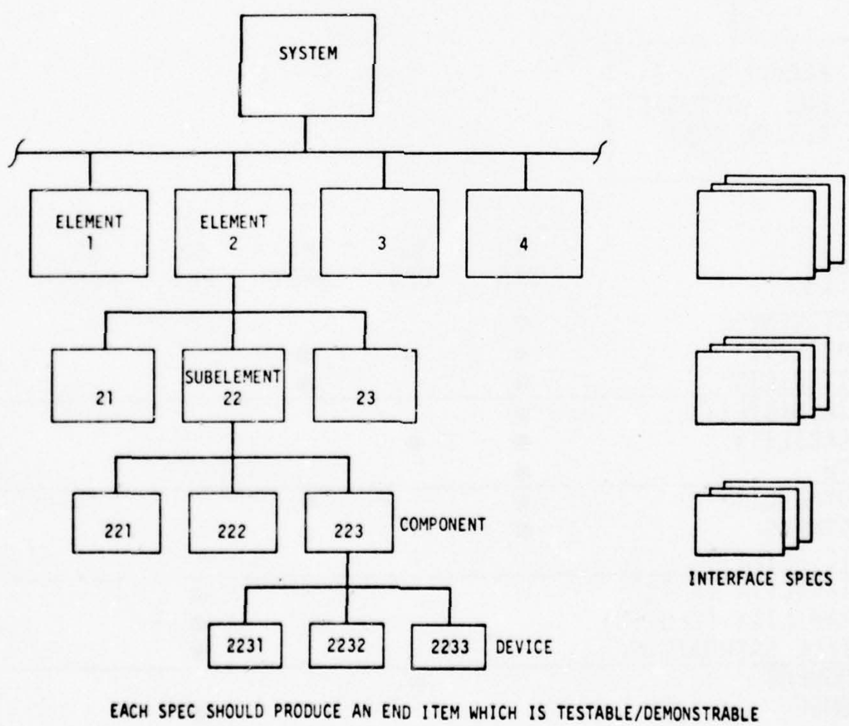


Figure 22. Specifications - Objectives Clearly Define Requirements for Accountability and Testability of Payoffs

SITE DEFENSE DP DESIGN APPROACH

Data processing capability and how it is implemented on the Site Defense program, are discussed in this subsection. The implementation technology in terms of hardware selection and software, is shown at the top of Table 9. The Site Defense approach was to use a top-down design and a large amount of simulation to verify the implementation. This included on the hardware selection such attributes as weighting and ranking, and benchmark testing. The software was based upon allocations to each of the engagement functions and then mapping into equations in order of solution which were implemented into code by structured programming techniques. The inputs and outputs were controlled in terms of an N^2-matrix which explicitly defines all of the inputs and outputs to each of the functioned elements. The algorithms in the Site Defense program were based on a number of different elements that are summarized as shown here. There were critical algorithm design studies, particularly in areas such as discrimination and tracking, where the design of the algorithm had a major impact on the system. Also, unit resource management algorithms were studied in great detail. Each algorithm that was of major importance and/or complexity had a simulation and simulation drivers developed for it. Finally, a task unit development folder was made which documented the progress, design and the problems of each of the algorithms or tasks as they were developed.

There were a number of formalisms used such as engagement logic. (Which explicitly defines all of the decisions and the performance factors for different decision levels, and explicitly defines all the possible logic combinations for the system that need to be implemented in the data processing). There were decision trees, such as leakage trees, that were put together to define performance levels; decision tables, which were used for lookup based on certain constraints and/or input values; convergence/stability analyses for resource management and other algorithms; threads and timing analysis were analyzed for a very large number of processing threads

TABLE 9. DATA-PROCESSING CAPABILITY

CAPABILITY	SITE DEFENSE APPROACH
• IMPLEMENTATION TECHNOLOGY	- TOP-DOWN DESIGN SIMULATION
- HARDWARE SELECTION	- ATTRIBUTES WEIGHTING/RANKING - BENCHMARK TESTS
- SOFTWARE	- FUNCTIONAL ALLOCATIONS - EQUATIONS IN ORDER OF SOLUTION - STRUCTURED PROGRAMMING - N^2 I/O MATRIX
• ALGORITHMS	- CRITICAL ALGORITHM DESIGN STUDIES - ALGORITHM SIMULATIONS AND DRIVERS - TASK UNIT DEVELOPMENT FOLDERS
• FORMALISMS	- ENGAGEMENT LOGIC - DECISION TREES - DECISION TABLES - CONVERGENCE/STABILITY ANALYSES - THREADS AND TIMING ANALYSIS - INSTRUCTIONS/TIMING BUDGETS - UNIT THROUGH PROCESS TESTING
• METHODS AND PROCEDURES	- SPECIFICATIONS - FUNCTION DESIGN NOTEBOOKS - PERT/SCHEDULE SYSTEM - INTERFACE WORKING MEETINGS - TEST PLANS AND PROCEDURES - OPERATING MANUALS - SYSTEM DEVELOPMENT FACILITY PLAN - KMR TEST AND MISSION PLANS - SOFTWARE DEVELOPMENT PLAN - SOFTWARE MANAGEMENT PLAN - QUALITY ASSURANCE PLAN - SOFTWARE STANDARDS AND PROCEDURES - CONFIGURATION MANAGEMENT - FORMAL REVIEWS - PERFORMANCE MEASUREMENT SYSTEM - COST CONTROL SYSTEM

to make sure that the connections through all the tasks were indeed accurate and complete, timing analysis was then done to determine the port-to-port timing to see if, under various load conditions, the timing budgets could be met. The instructions and timing budgets were allocated in a formal manner to each of the functional areas for the purpose of allocating performance and trading off the amount of real-time execution versus the amount of complexity and performance levels to be achieved. Also, testing was done from unit level through process level in a formal manner. The methods and procedures used are listed in the table.

Each of the major function requirements had a separate function design notebook which had about 15 or 20 areas of the design. Such items as performance levels, input requirements to the function, key algorithms, test data, references, etc., and logic diagrams were included as well. A PERT and scheduling system was used for all of the developments and requirements; interfacing working meetings were held numerous times to make sure that the interfaces among the functions were valid; test plans and procedures were generated and operating manuals were prepared.

A system development facility plan was prepared along with KMR test and mission plans, a software development plan and software management plan. These plans set policy and guidelines for the developments. The quality assurance plan included all of the QA provisions. A software standards and procedures document was used along with configuration management, including a configuration patrol board. Formal reviews for the design were held, a performance measurement system was used for a time and a cost control system was followed. These various methods and procedures for the most part are self-explanatory and each of them in themselves could be the subject of a separate briefing. In summary the Site Defense system was developed using a well-balanced set of methods and procedures that produced the quality product that was finally achieved. Although the learning process caused some starting and stopping on certain of these approaches,

in general, those which are cited on the approach column all produced useful results in an efficient manner. There was no single method or procedure that could be used as a panacea for the entire development, but rather the appropriate tools had to be used in the appropriate place to gain the appropriate information. It requires the skillful knowledge and use of people developing the product, for these techniques to be successful.

SITE DEFENSE DATA PROCESSING SUMMARY

A data-processing summary for the Systems Technology Program, Site Defense and Hardsite is illustrated in Figure 23. The data-processing subsystem consists of hardware and software. The CDC 7700 is the hardware that was selected. A commercial data processor was required and the characteristics of the CDC 7700 are shown on the figure. The software included tactical application programs, tactical operating systems and a pre-engagement program for the principal part of application and engagement software. This is about 170K instructions. The operational and support software was approximately twice that size and the development test support software which was non-real time, was approximately 600K instructions. These software packages were developed in a series of phases which were called at various times loops or steps and each of these various loops produced a certain level of capability which was incrementally increased until the final desired capability was achieved.

The figure at the bottom of the chart shows an architecture of the CDC 7700 with a large core memory of 512K words which contains all of the data base and tactical operating systems controls, two CPUs, one on each side, as shown in the figure. Each of the CPUs has a small core memory of 65K words. There are I/0 MUXs that connect to the peripheral processing units on each side. Specific details of the 7700 can be obtained from the manuals.

SYSTEMS TECHNOLOGY PROGRAM/SITE DEFENSE/HARDSITE

DATA PROCESSING SUBSYSTEM

- HARDWARE - CDC 7700

  CLOCK PERIOD 27.5 ns
  LCM ACCESS 275 ns
  64 INSTRUCTIONS SET
  CORE STORAGE 512,000 WORDS
  INPUT/OUTPUT 300 × 10^6 BITS/s
  CPU PARTITIONED IN 9 STEPS
  THROUGHPUT 20 TO 26 MIPS

- SOFTWARE

TACTICAL APPLICATION PROGRAMS TACTICAL OPERATING SYSTEM PRE-ENGAGEMENT PROGRAM	} 170K INSTRUCTIONS
OPERATIONS & SUPPORT SOFTWARE	} 300K INSTRUCTIONS
DEVELOPMENT & TEST SUPPORT SOFTWARE	} 600K INSTRUCTIONS

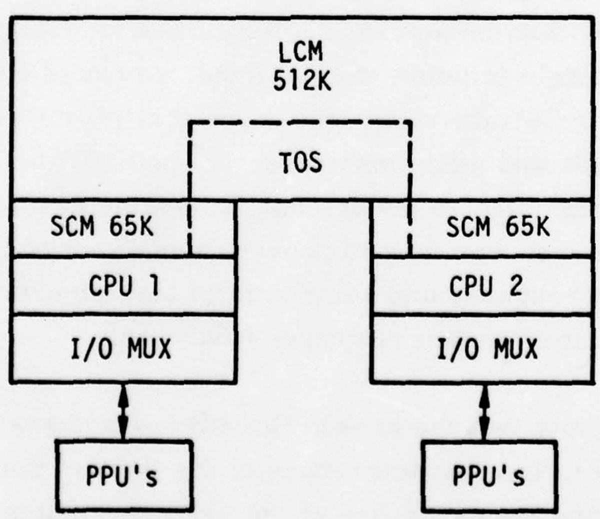


Figure 23. Data-Processing Summary

SITE DEFENSE INDICATED RESEARCH

A review of BMD systems identified several key areas that were indicated for research: Table 10 shows particular areas of research that were based upon observation from the Site Defense System. Basically, the system meets its baseline requirements and the indicated research then would tend to lie in areas for advanced cases which may exceed the current design -- the cases for advanced threats for additional missions that are beyond the primary objectives of the Site Defense System. Additionally, these advanced cases could apply to situations which would be more cost effective than the baseline design. For example, even with the same mission it may be more cost effective to go to distributed data processing than to use the large mainframe 7700. The baseline design does meet the baseline requirements in general. However, some port-to-port times are not achievable. There are certain port-to-port timing requirements that are faster than 25 ms which are very difficult to meet at high probability levels, particularly under any type of load situation. This is due to the hardware contention and the software design. The hardware contention for small-core memory results from the fact that only a single task can be executed at a time, and it has to queue up and wait until its turn. Furthermore, the threads are designed within the software to require a multiplicity of tasks to be executed before a single impulse is completed, so based on wait times and processing times for all the tasks on a given thread, as well as the contention for other threads and other instances, the bottom line is that any port-to-port times faster than 25 ms are not achievable at high probability levels. Indicated research would be to provide some sort of local processing for a high, a very fast response and some sort of task structure redesign technique that would allow for fast response situations.

The third observation was that the growth flexibility for threat evolution is somewhat brittle, due to the characteristics of the threat, the penaids and tactics of the threat, and the capacities of the system. Indicated research

TABLE 10. REVIEW OF BMD SYSTEMS IDENTIFIED SEVERAL KEY AREAS FOR RESEARCH--SITE DEFENSE SYSTEM

OBSERVATION	INDICATED RESEARCH
• BASICALLY MEETS BASELINE REQUIREMENTS	• ADVANCED CASES WHICH EXCEED DESIGN
• SOME PORT TO PORT TIMES NOT ACHIEVABLE - FASTER THAN 25 ms - HARDWARE CONTENTION - SOFTWARE DESIGN	• LOCAL PROCESSING FOR HIGH RESPONSE; TASK STRUCTURE REDESIGN
• GROWTH FLEXIBILITY FOR THREAT EVOLUTION SOMEWHAT BRITTLE - CHARACTERISTICS - PENAIDS AND TACTICS - CAPACITIES	• DESIGN TECHNIQUES TO ASSURE FLEXIBILITY
• DATA ACCESS COSTLY - MAJOR PART OF TACTICAL OPERATING SYSTEM (TOS) COST - TOS CONSUMES OVER 50% OF PROCESSOR UTILIZATION AT STRESSING POINTS	• MORE EFFICIENT DATA BASE FILE MANAGEMENT
• LIMITED CAPABILITY DESIGNED-IN FOR CASUALTY/DEGRADED DATA PROCESSING OPERATION	• DEGRADED CAPABILITY (RE)CONFIGURATIONS AND TECHNIQUES
• ERROR DETECTION CODE COSTLY - RADAR SCHEDULING TASK INSTRUCTIONS INCREASED ABOUT 10%	• MORE EFFICIENT ERROR DETECTION AND CORRECTION TECHNIQUES
• PERFORMANCE VALIDATION DIFFICULT - REQUIREMENTS TESTABILITY ISSUE - NUCLEAR EFFECTS ISSUE	• DESIGNED-IN TESTABILITY METHODS; TESTING TOOLS INNOVATIONS
• RESOURCE MANAGEMENT ALGORITHMS DIFFICULT	• SIMPLER INTERFACES, CONSTRAINTS, ALGORITHMS
• TASKS MAY BE GROUPED BY DOMINANT PROCESSING TYPE - LOGICAL - MATHEMATICAL (e.g., OBJECT TRACK)	• SPECIAL HARDWARE/FIRMWARE/SOFTWARE ARCHITECTURES

is that design techniques to assure flexibility would be desirable. Before the software can be implemented specific decision tables have to be generated, specific performance curves have to be prepared, and as a result of these and the dependency on threat characteristics, a certain amount of brittleness becomes embedded into the design. For example, the two major classes of threats in the baseline system in the early days included a large threat and a small threat. Based on the discrimination decision, the intersect planning and intersect control functions would make use of that information in flying the missile out to the intersect point. If the decision was wrong there was a higher probability that the target would be missed. This indicates some brittleness in terms of design. Furthermore, if the threats that were engaged were not one of the two baseline types, but rather some off-nominal point, it is not clear that the basic intercept tactics would work for all situations. Consequently, design techniques need to be built or developed to assure that flexibility can be included.

The fourth observation is that data access is very costly. The major part of the tactical operating cost was due to data access and data servicing. The tactical operating system consumes over half of the processor utilization at load, stress and points. Indicated research is that more efficient data base file management is needed. As the data access costs become large the efficiency of the system will go down. As distributed processing takes place and the size of the data base can be reduced to local subsets, it appears that the data access would be somewhat reduced. This has yet to be proven and should be a topic of the research and the followon study. The next observation is that limited capability was designed in for casualty or degraded data-processing operation (alluded to in an earlier figure), and the indicated research is that degraded capability reconfigurations and techniques should be examined and laid out. Particularly with the newer technology this could be a very cost-effective alternative. The error detection code was very costly and, for example, on the radar scheduling task the instructions increased at least 10 percent strictly for error detection. More efficient

error detection and correction techniques are probably available and could be implemented with solutions other than purely software.

Next, the performance validation is very difficult, primarily due to requirements testability issues and also phenomenology issues such as nuclear effects. To address the first part, design and testability methods are essential. To allow one to verify that the requirements are indeed met, the testability has to be designed in. Innovations in testing tools are needed for nuclear effects. For example, it will not be possible to really verify that the modeling and responses are accurate. Resource management algorithms are very difficult. The indicated research is simpler interfaces, constraints, and algorithms; simpler techniques that can be implemented. Finally, tasks may be grouped by dominant processing type. This is an important statement with respect to DDP since certain of the tasks are predominantly logical tasks and certain are predominantly mathematical tasks, so based upon the types of task compositions, one should be able to devise special hardware, firmware or software architectures that take advantage of that. For example, the object tracking algorithms processors or Kalman filter chips that could process many objects simultaneously. (The use of PEPE is also being looked at for this purpose.) The logical-versus-mathematical design is one that lends itself to particular system set of functions and then maps directly into requirements on the data processing implementations. The tradeoff that needs to be worked between the software, firmware, and hardware is one which is based upon the set of requirements, in terms of timing and type of processing indicated.

RESEARCH ISSUES: OVERVIEW

Table 11 identifies some selected research topics which lead to payoffs in key attribute categories. Fifteen BMD requirements issues shown in the table are mapped over to the performance, cost, schedule and risk attributes categories. Each of these 15 items has been based upon observations from BMD requirements and issues that exist in requirements that would indicate further research is needed.

TABLE 11. SUGGESTED RESEARCH TOPICS LEAD TO PAYOFFS IN KEY ATTRIBUTE CATEGORIES

	PERFORMANCE	COST	SCHEDULE	RISK
• BMD REQUIREMENTS ISSUES:				
(1) ALLOCATION AND SCHEDULING OF BMD RESOURCES	✓			
(2) ALLOCATION AND SCHEDULING OF DP RESOURCES	✓			
(3) CONCEPT SPECIFIC ALGORITHM FEASIBILITY	✓			✓
(4) DECENTRALIZED "GLOBAL" FUNCTION FEASIBILITY	✓			✓
(5) FAILURE MANAGEMENT ALGORITHM FEASIBILITY	✓			✓
(6) COMMUNICATION NETWORK TOPOLOGIES	✓	✓	✓	
(7) HARDWARE AND SOFTWARE REDUNDANCY	✓			✓
(8) HIGH RELIABILITY COMPONENTS	✓	✓		✓
(9) FAULT TOLERANCE	✓	✓		✓
(10) INDEPENDENT VALIDATION OF COMPONENTS		✓		✓
(11) PERFORMANCE VALIDATION OF COMPOSITION RULES		✓	✓	✓
(12) HARDWARE TEST BED		✓	✓	✓
(13) SOFTWARE DEVELOPMENT SYSTEM		✓	✓	✓
(14) DATA BASE DEFINITION			✓	
(15) MAINTENANCE OF DATA BASE CONSISTENCY	✓			✓

Similar observations were made from BMD experience and also DP technology issues (Table 12). The listing of these research topics is intended to be a summary statement which is based upon the previous material that was presented and also the experience and observations derived from Site Defense in particular. Further, the data-processing technology issues are derived from the state of the art in terms of publications, developments, and devices that are contemporary. It is the task of a subsequent study to firm up the details and specific research objectives of each of these topics. As an example of the next level of detail, two categories have been selected. First is item number 9 on Table 11, which is fault tolerance; and second is combined items topics 14 and 15, data base definition and maintenance of data base consistency.

TABLE 12. SUGGESTED RESEARCH TOPICS LEAD TO PAYOFFS IN KEY ATTRIBUTE CATEGORIES (CONTINUED)

	PERFORMANCE	COST	SCHEDULE	RISK
• BMD EXPERIENCE ISSUES:				
(1) LOCAL PROCESSING FOR HIGH RESPONSE	✓			✓
(2) DESIGN TECHNIQUES TO ASSURE FLEXIBILITY		✓	✓	✓
(3) MORE EFFICIENT DATA BASE MANAGEMENT	✓			
(4) COST EFFECTIVE DEGRADED CAPABILITY CONFIGURATIONS AND TECHNIQUES	✓	✓		✓
(5) MORE EFFICIENT ERROR DETECTION AND CORRECTION TECHNIQUES	✓			✓
(6) DESIGNED-IN TESTING METHODS		✓		✓
(7) SIMPLER INTERFACES AND ALGORITHMS		✓		✓
(8) SPECIAL HARDWARE/SOFTWARE/FIRMWARE ARCHITECTURES	✓	✓	✓	✓
• DP TECHNOLOGY ISSUES:				
(1) LARGE MEMORY TECHNOLOGY	✓			
(2) FIRMWARE ADAPTATION TO DDP TASKS	✓	✓	✓	
(3) RESOURCE MANAGEMENT IN A DISTRIBUTED ENVIRONMENT	✓			✓
(4) DISTRIBUTED DATA BASE IMPLEMENTATION	✓	✓	✓	
(5) SOFTWARE ENGINEERING FOR DISTRIBUTED HARDWARE		✓	✓	✓
(6) HARDWARE SWITCHING TECHNIQUES	✓			✓
(7) SOFTWARE SWITCHING TECHNIQUES	✓			✓
(8) COMMUNICATIONS MESSAGE HANDLING IN DDP	✓			
(9) SOFTWARE MAINTENANCE IN A DISPERSED SYSTEM	✓	✓		✓
(10) HARDWARE MAINTENANCE IN A DISPERSED SYSTEM		✓		
(11) HARDWARE AND SOFTWARE ANALYSIS TOOLS		✓	✓	

RESEARCH ISSUE: FAULT TOLERANCE

Table 13 shows the research topic for development of a fault-tolerant, large-scale integrated circuit application, a potential high-payoff research topic. The objective first is to define the application of LSI devices to achieve data processor fault tolerance, and second to develop fault-tolerant concepts to enable scheduled maintenance and self-test and repair capability. The approach is given in five steps: 1) select fault tolerant approaches for possible application; 2) identify LSI devices which can be used to implement fault tolerance; 3) perform analysis for specific applications; 4) develop prototypes using fault tolerant LSI memories and processors; and 5) review the applications to BMD data-processing subsystems. This approach and these steps would then serve to provide an input to BMD problems and BMD data-processing development. The analysis alluded to here would be in the form of numerical quantities, calculations for specific designs and specific requirements.

RESEARCH ISSUE: DISTRIBUTED DATA BASE

The development of distributed data base design and methodology is a critical research topic in the distributed data processing approach (Table 14). The objectives would be to define a data base structure which can be mapped onto a flexible network of processing nodes and to develop the methodology required to maintain the data base which can be distributed to the nodes by function; identify the interfaces required to maintain the data base; perform analysis of information flow between elements of the distributed data base, including the concepts of control, updating and editing of the data; minimize information flow; and, finally, test the subsets for validity. The purpose of distributed data base, of course, is to minimize the data flow and to simplify the processing done on a given node.

TABLE 13. RESEARCH TOPIC--FAULT TOLERANCE

- DEVELOPMENT OF A FAULT TOLERANT LARGE SCALE INTEGRATED CIRCUIT APPLICATION IS A POTENTIAL HIGH PAYOFF RESEARCH TOPIC

OBJECTIVES

- DEFINE THE APPLICATION OF LSI DEVICES TO ACHIEVE DATA PROCESSOR FAULT TOLERANCE
- DEVELOP FAULT TOLERANT CONCEPT TO ENABLE
  - SCHEDULED MAINTENANCE
  - SELF TEST AND REPAIR CAPABILITY

APPROACH

- SELECT FAULT TOLERANT APPROACHES FOR POSSIBLE APPLICATION
- IDENTIFY LSI DEVICES WHICH CAN BE USED TO IMPLEMENT FAULT TOLERANCE
- PERFORM ANALYSIS FOR SPECIFIC APPLICATIONS
- DEVELOP PROTOTYPES USING FAULT TOLERANT LSI MEMORIES AND PROCESSORS
- REVIEW APPLICATIONS TO BMD DATA PROCESSING SUBSYSTEMS

TABLE 14. RESEARCH TOPIC--DISTRIBUTED DATA BASE

- DEVELOPMENT OF DISTRIBUTED DATA BASE DESIGN AND METHODOLOGY IS A CRITICAL RESEARCH TOPIC IN THE DISTRIBUTED DATA PROCESSING APPROACH

OBJECTIVES

- DEFINE A DATA BASE STRUCTURE WHICH CAN BE MAPPED ONTO A FLEXIBLE NETWORK OF PROCESSING NODES
- DEVELOP THE METHODOLOGY REQUIRED TO MAINTAIN THE DATA BASE DURING AN ENGAGEMENT

APPROACH

- IDENTIFY SUBSETS OF THE DATA BASE WHICH CAN BE DISTRIBUTED TO THE NODES BY FUNCTION
- IDENTIFY THE INTERFACES REQUIRED TO MAINTAIN THE DATA BASE
- PERFORM ANALYSIS OF INFORMATION FLOW BETWEEN ELEMENTS OF THE DISTRIBUTED DATA BASE
- MINIMIZE INFORMATION FLOW WHERE POSSIBLE
- TEST SUBSETS FOR VALIDITY

IC TECHNOLOGY

Table 15 addresses IC technology. Large-scale integrated circuits are evolving with characteristics appropriate for application to BMD. Additional research is required to develop applications of specific devices to BMD problems and to steer the industrial R&D. For example, the industrial R&D at the current point in time is tending to lead the military complex, as opposed to what happened 10 or 15 years ago when the government tended to drive all the major developments. Consequently, this puts the government in the situation of following and trying to apply industrial R&D, industrial products. The trends for technology are shown on the figure. In 1977 we have a computer on a chip which has high speed, high-to-low temperature range and very low power. By 1980 the speed will be even higher and radiation hardening will begin to be implemented; the testability on the chip will begin to be implemented. By the early 1980s a higher complexity will be available, including higher speeds, lower power, radiation hardening, and self-repair, which would produce basically a system on a chip which has perhaps 10,000 gates. The key issue, then, is to reduce the risk of performance, cost, and schedule sufficient for applications, so that these can be used. These general trends then are expected and should be followed and applied to the BMD problem.

TABLE 15. IC TECHNOLOGY

- LARGE SCALE INTEGRATED CIRCUITS ARE EVOLVING WITH CHARACTERISTICS APPROPRIATE FOR APPLICATION TO BMD
- ADDITIONAL RESEARCH IS REQUIRED TO DEVELOP APPLICATIONS OF SPECIFIC DEVICES TO BMD PROBLEMS AND TO STEER INDUSTRIAL R&D

TECHNOLOGY TRENDS

1977	1980	1982/4
	HIGHER SPEED	HIGHER COMPLEXITY
HIGH SPEED	HI/LOW TEMP	HIGHER SPEED
HI/LOW TEMP	LOWER POWER	HI/LOW TEMP
LOW POWER	RADIATION HARD	LOWER POWER
	TESTABILITY	RADIATION HARD
		TESTABILITY
		SELF REPAIR
COMPUTER ON-A-CHIP		SYSTEM ON-A-CHIP

- KEY ISSUE IS TO REDUCE RISK OF PERFORMANCE, COST, AND SCHEDULE SUFFICIENT FOR APPLICATIONS

BIBLIOGRAPHY

Parallel Processor for Augmenting the ARTS III System -- Transportation System Center of the Department of Transportation, Contract No. DOT-TSC-625, June 1973 - December 1973.

All-Semiconductor Distributed Aerospace Processor/Memory (DP/M) Study -- Air Force Systems Command (AFSC), Contract No. F33615-72-C-1709, Project No. 2003, May 1972 - August 1973.

Feasibility Demonstration of Distributed Processing for Small Ships Command and Control (SSCC) Systems -- Naval Electronic Laboratory Center (NELC), Contract No. N00123-74-C-0891, December 1973 - May 1976.

Evaluation of the Use of an Associative Processor in Communication Multiplexing -- Rome Air Development Center (RADC), Contract No. F30602-71-C-0350, July 1971 - August 1972.

Associative Buffer Concentrator (ABC) -- RADC, Contract No. F30602-74-C-0042, November 1973 - October 1975.

Parallel-Element Processing Ensemble (PEPE) Implementation Study -- A subcontract to System Development Corporation for the U. S. Army Advanced Ballistic Missile Defense Agency (ABMDA), Subcontract No. SDC-71-61, January 1971 - July 1971.

Prototype Bulk-Filter Development Concept Definition -- ABMDA, Contract No. DAH 60-72-C-0052, December 1971 - March 1972.